Task-level run-time scheduling approach for dynamic real-time multi-media systems

Francky Catthoor, IMEC, Belgium

Current TCM research project members:

IMEC: Pol Marchal, Chun Wong, Peng Yang

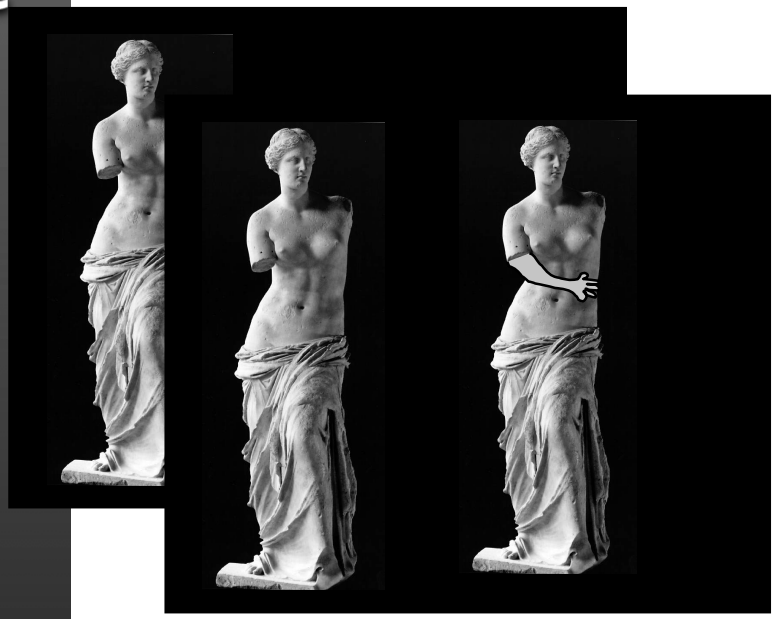
K.U.Leuven-ESAT: Stefaan Himpe

Goal of our research

- A methodology to map *dynamic* and *concurrent* real-time applications on an embedded multi-processor platform

Why are Applications becoming more dynamic and concurrent?

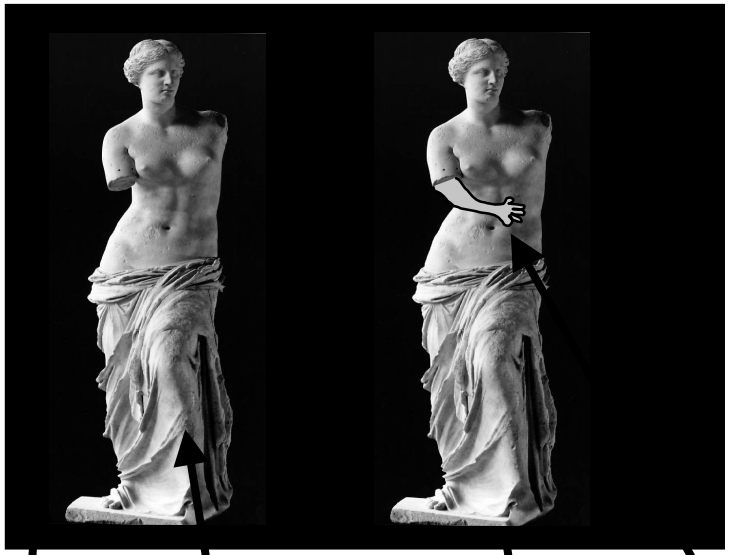
JPEG



T1

T2

MPEG4



T1'

T1

T3

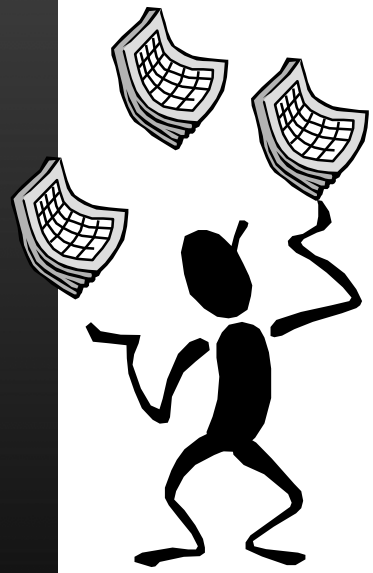
T4

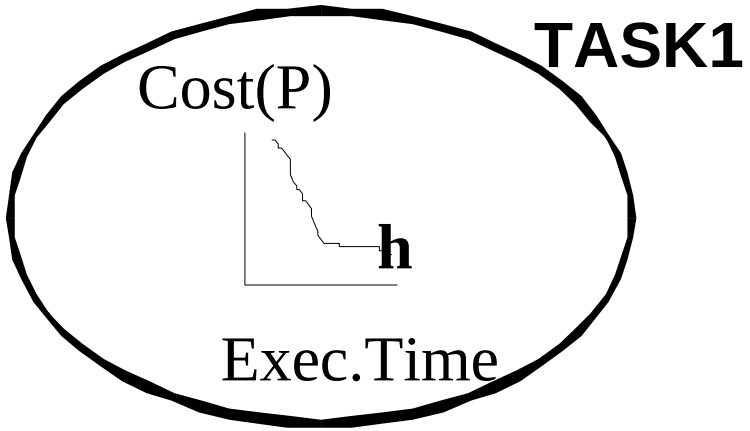
The workload decreases but the tasks are dynamically created and their size is data dependent

Local picture

MPEG-4 : multimedia spec
= too huge to handle as 1 task
=> break up in many interacting tasks

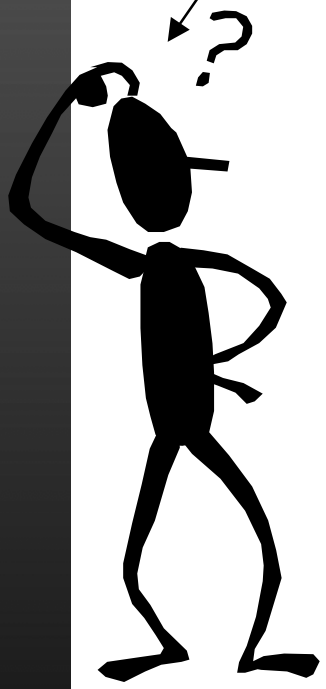
With this code my boss has to give me a raise!!!

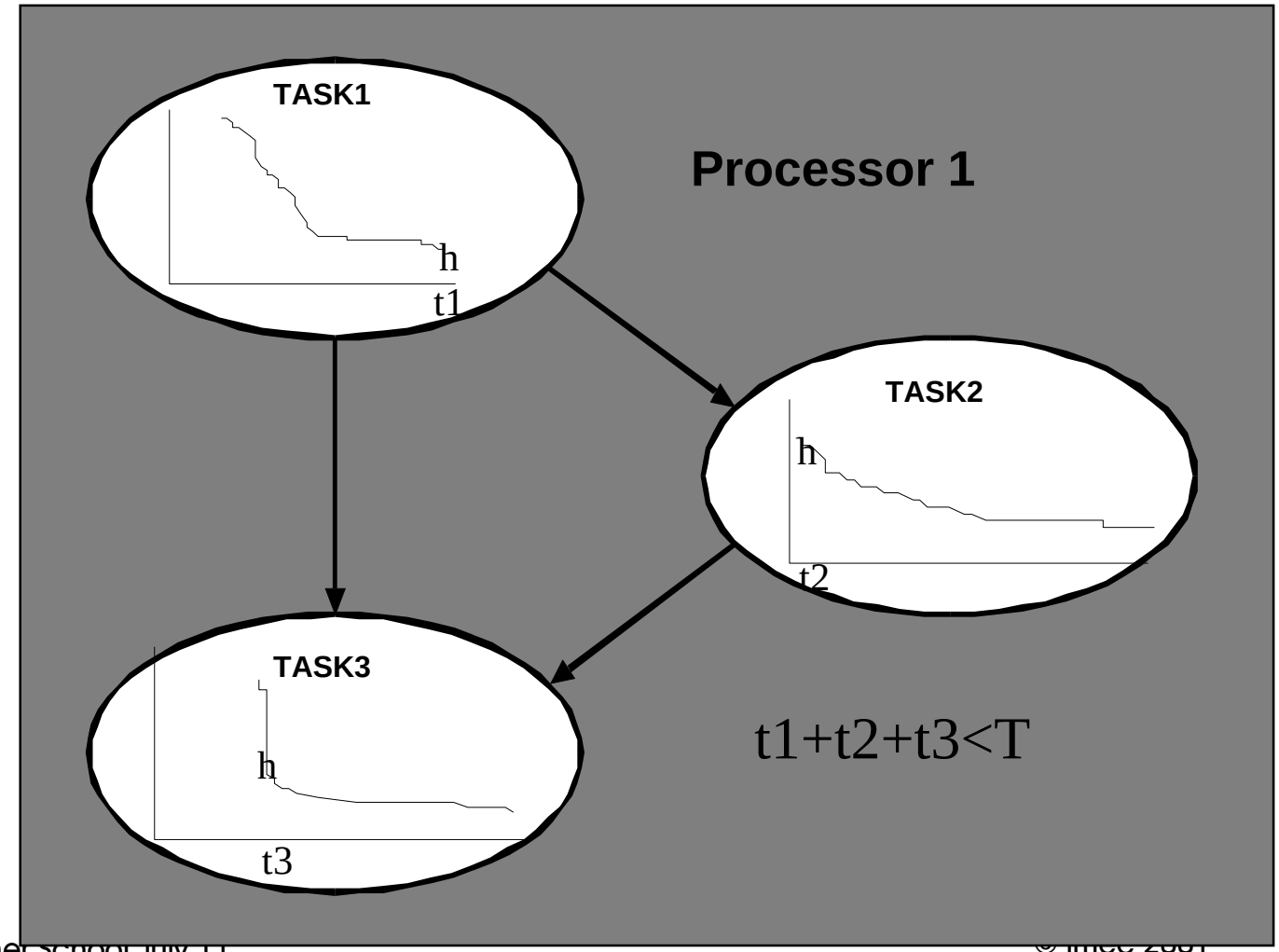




Global picture: ad hoc

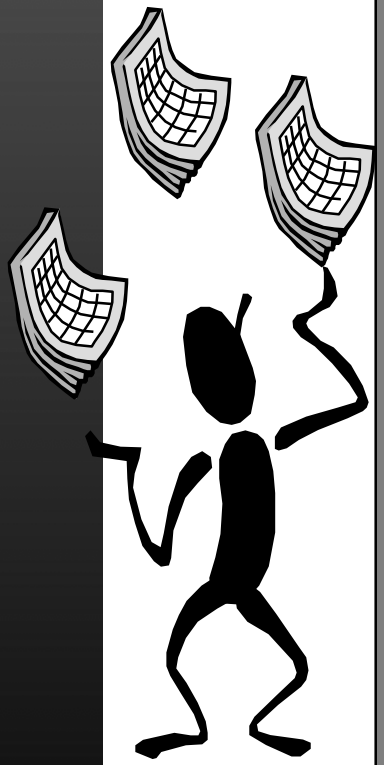
This didn't look so good after all???

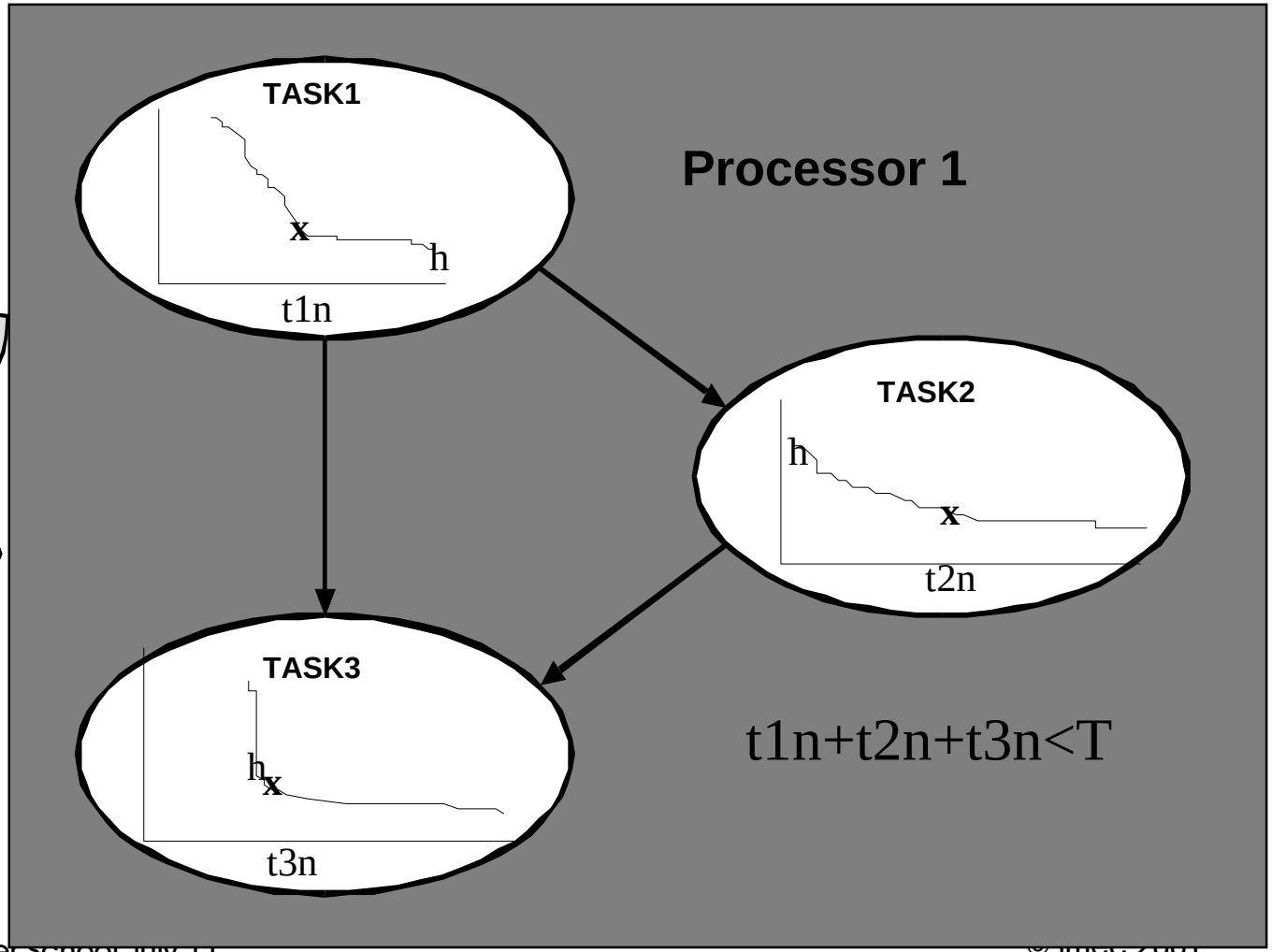




Global trade-offs with cost-performance curves

Luckily we have the Pareto approach!

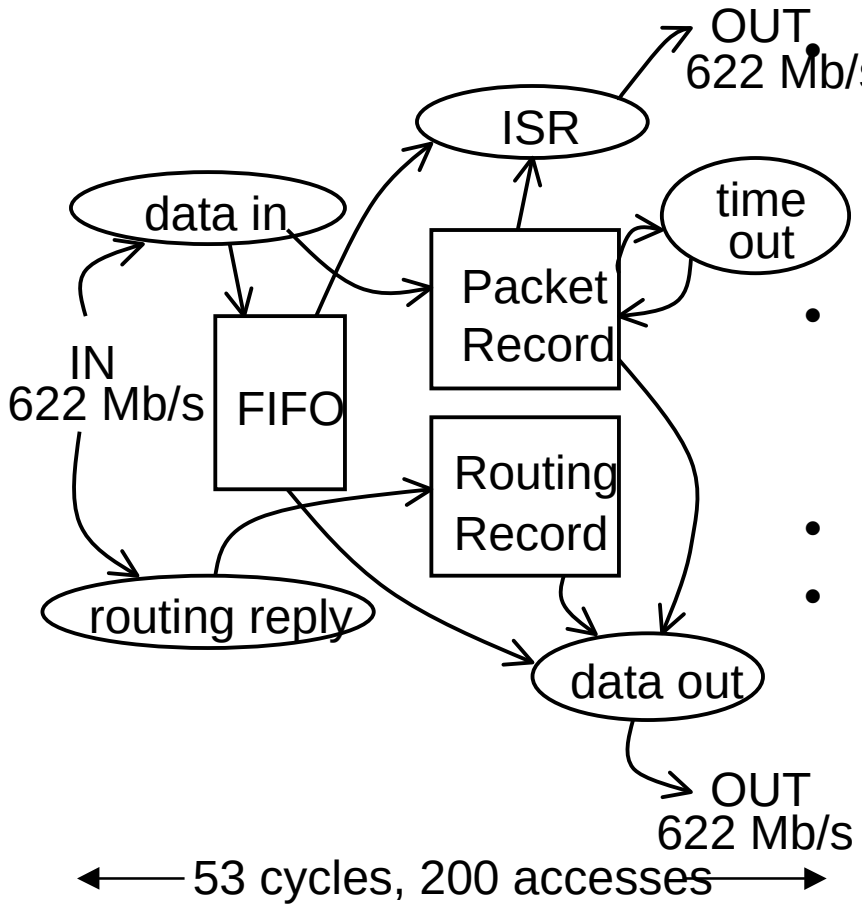




Outline

- *Motivation: challenges in the system-level design*
- Overview of methodology
- Cost-efficient run-time scheduling for RTOS
- Long term research challenges

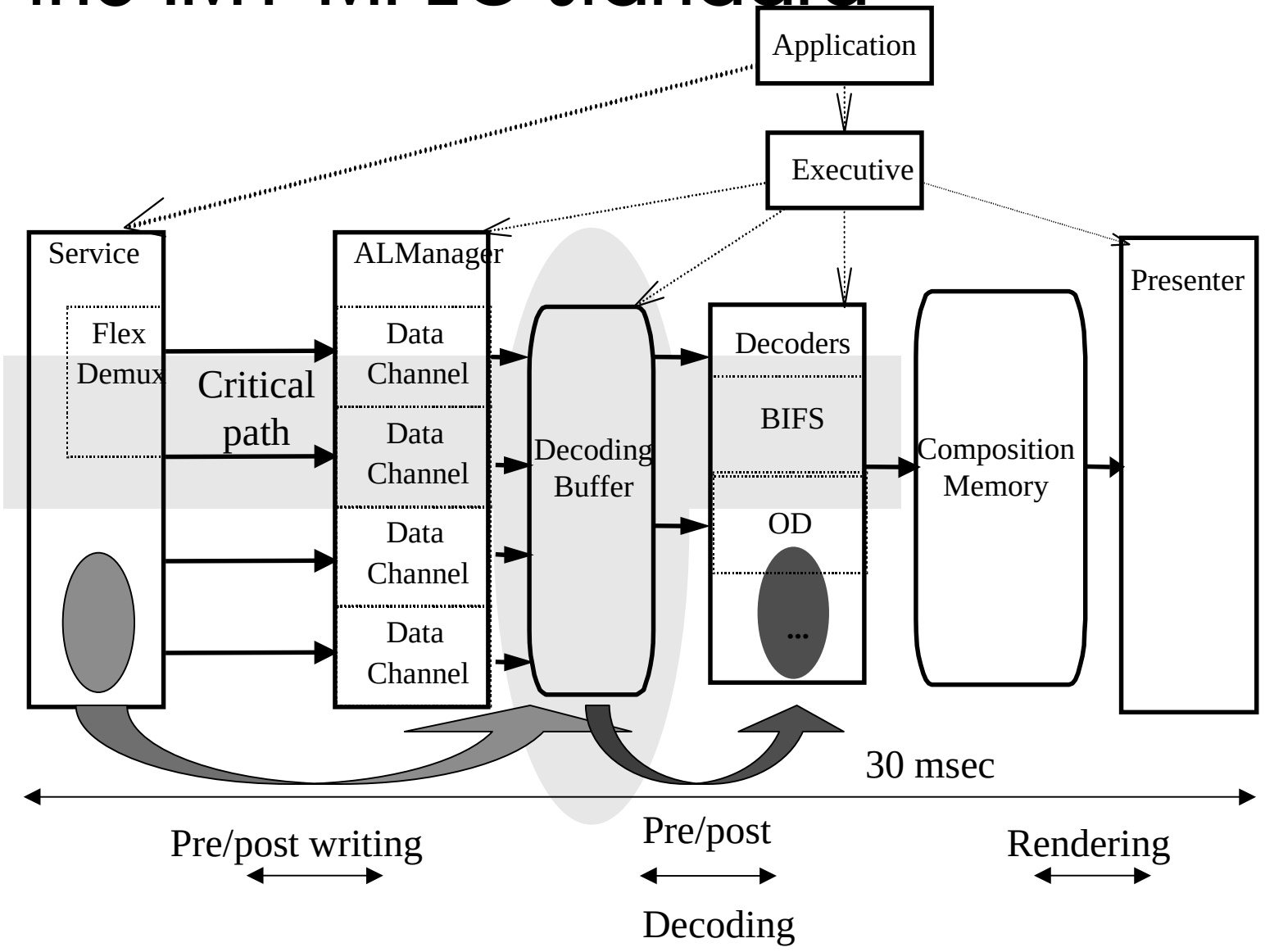
Application domain



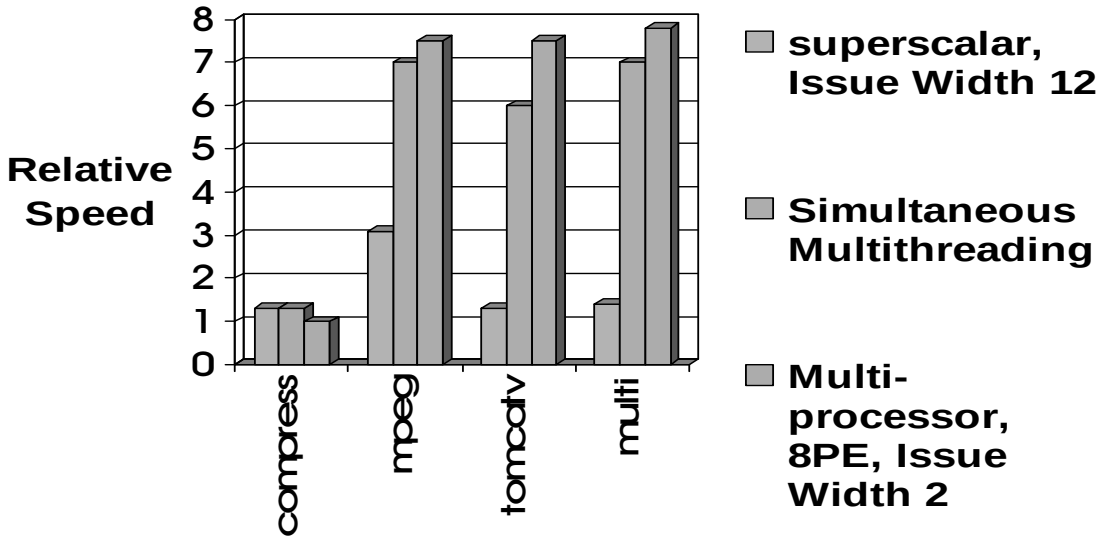
- Processes
  - Dynamic and concurrent processes
  - Global/local control
  - Little data processing
- Complex data sets
  - Large and irregular dynamically allocated data
  - Huge memory accesses
- Stringent real-time constraints
- Embedded system

E.g.: Network layer protocols (ATM, IP, …)
Multi-media algorithms with dynamic character (MPEG4, MPEG7)
Wireless and wired terminals (Internet, WLAN, ADSL, …)

Real-time constraints and timing in the IM1-MPEG standard



Target architecture: Chip Multi Processor (CMP)

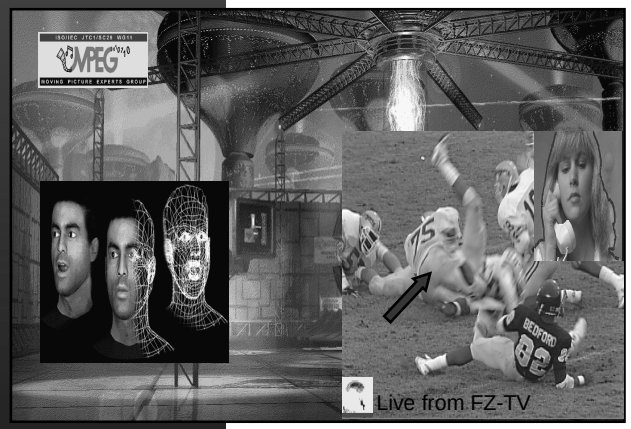


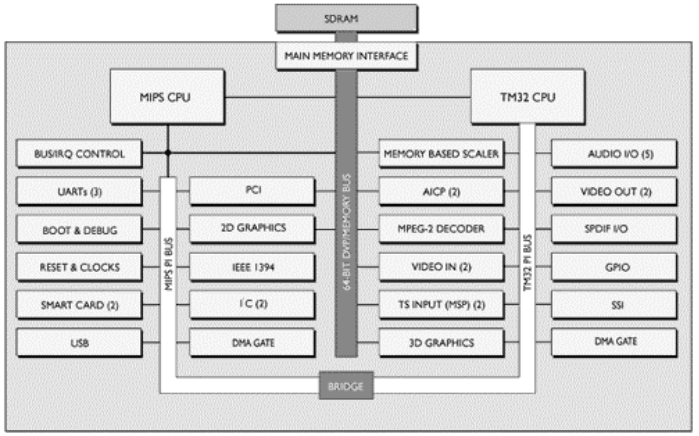
From L. Hammond, IEEE Computer, Sept 1997

- Advantages:
  - Performance: possibility to exploit thread level parallelism combined with ILP
  - Energy: low energy cost per instruction by customizing the nodes (ASIPs) + effective memory hierarchy and distributed customisable organisation
  - Flexible: programmable nodes
  - Scalability: memory bandwidth is scalable (if good memory hierarchy is used)

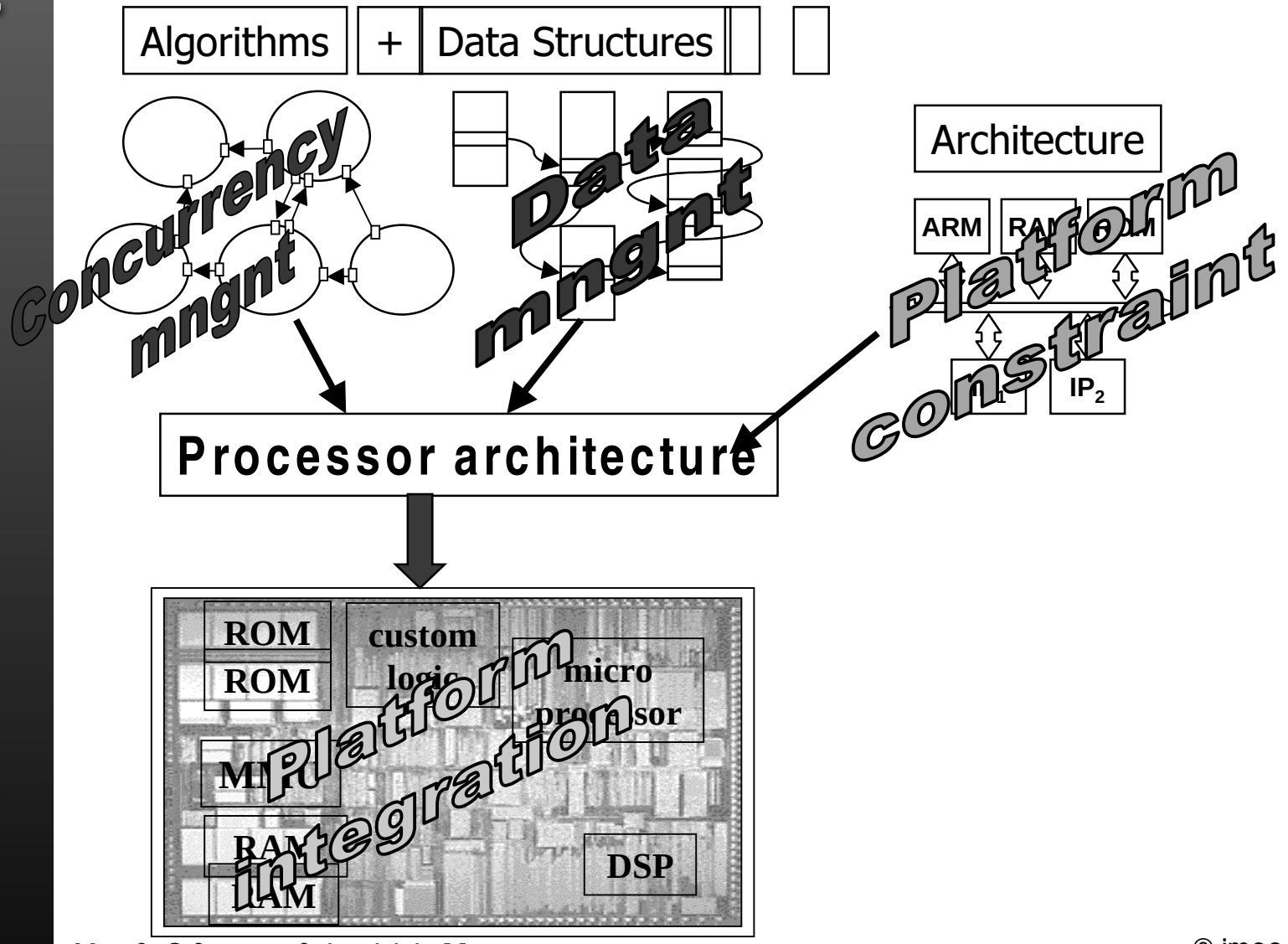
Why aren't CMPs used now?

- Multi processors are used in servers and scientific computing
- Not in the context of embedded systems
  - efficient mapping requires a very high design effort when done manually
  - need for a compiler





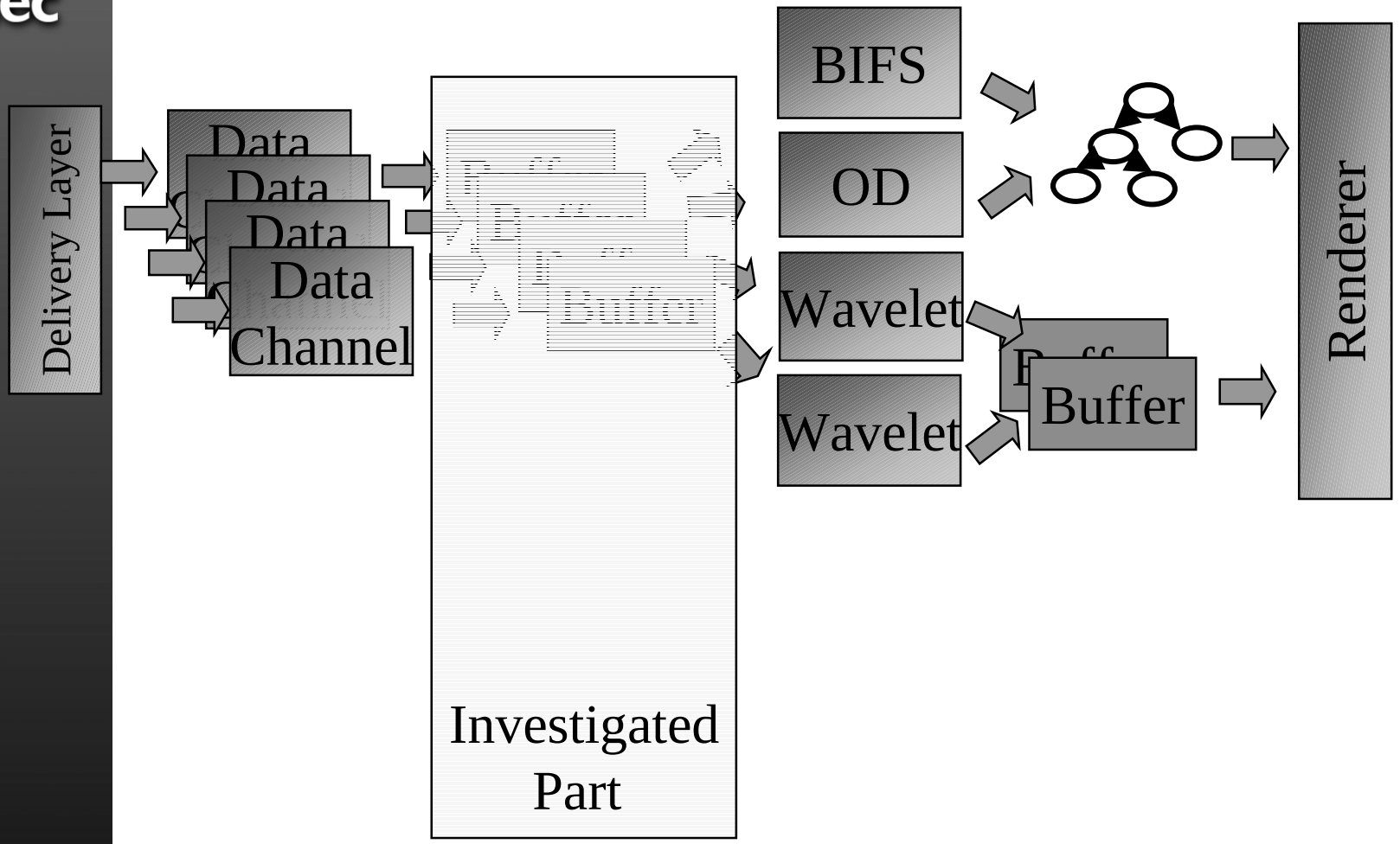
C/C++ system refinement + exploration



Outline

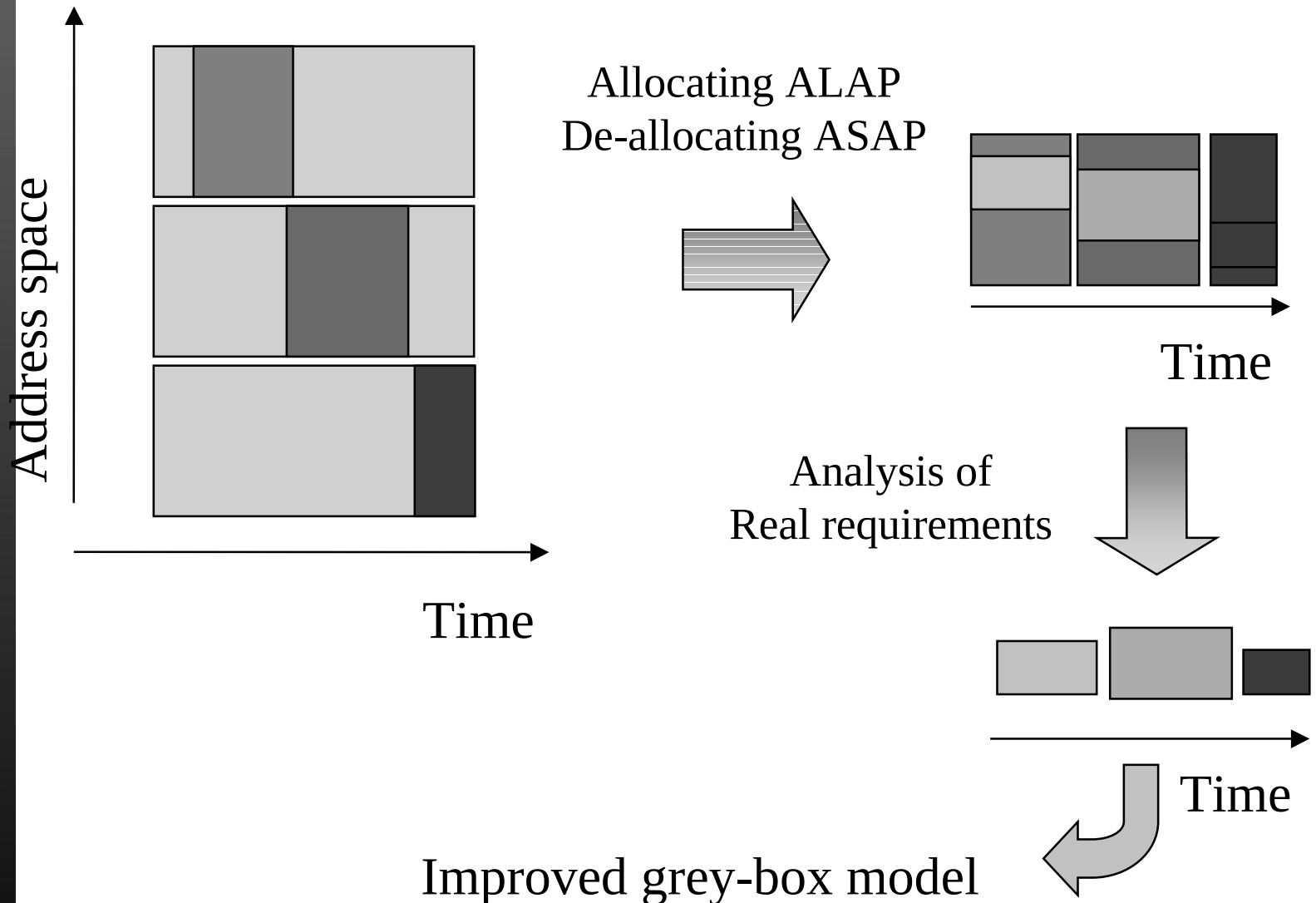
- Motivation: challenges in the system-level design
- *Overview of methodology: data management*
- Cost-efficient run-time scheduling for RTOS
- Long term research challenges

Task Level DTSE on the IM1-player

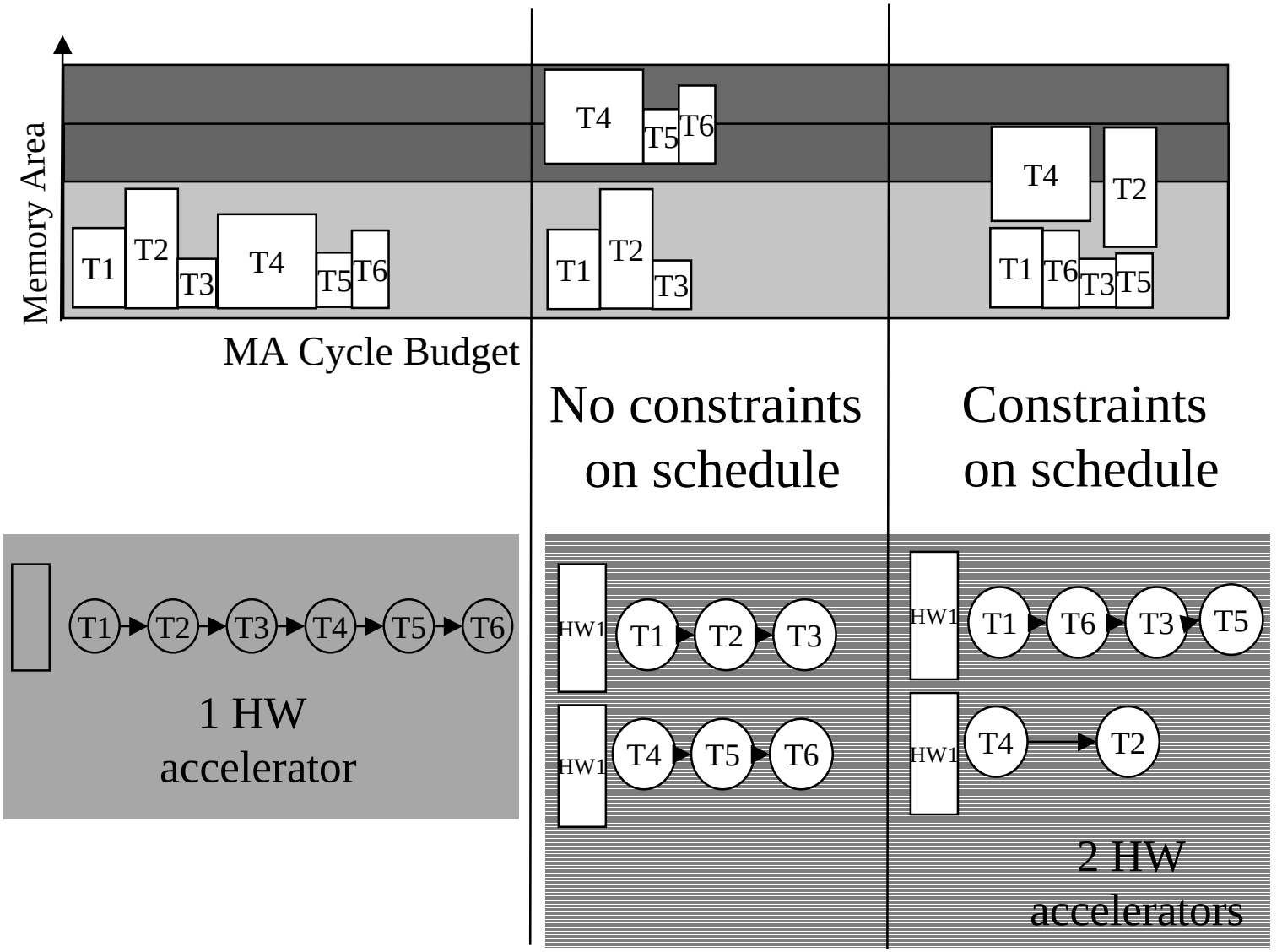


PACT'00 (COLPwsh), PACS'00

Platform Independent Code Transformations

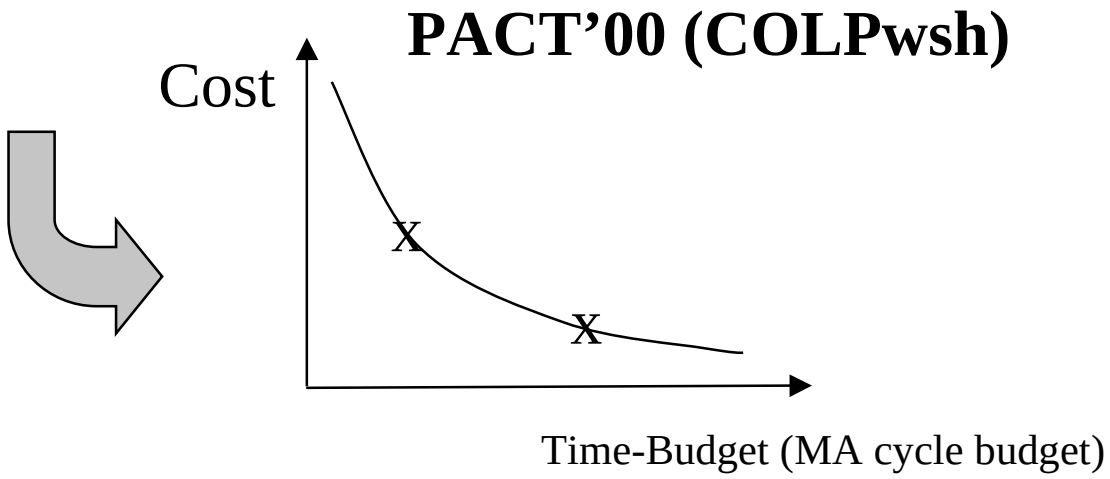


Access ordering and generation of the task (partial) precedence constraints



Results on IM1 player

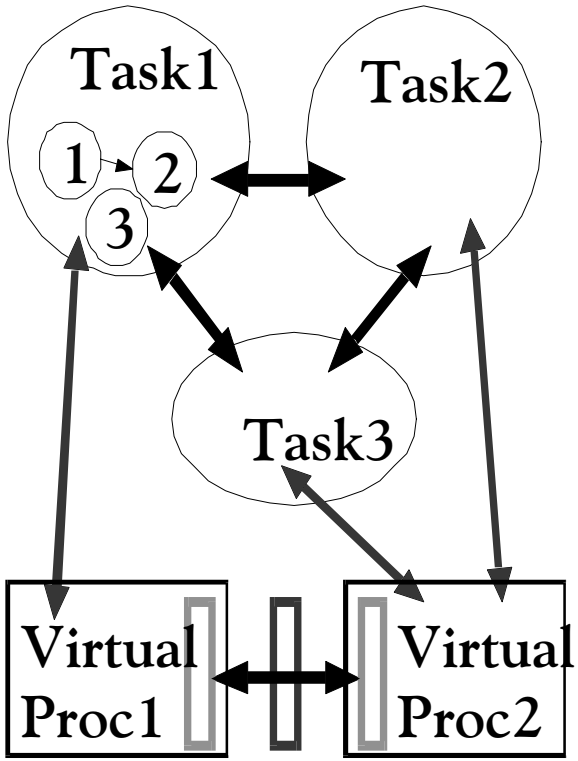
	Memory Size Pre	*Memory Size Post*	*Memory Energy Pre*	*Memory Energy Post*
1Proc	86.9kB	14 .8kB	0.78mJ	0.16mJ
2Proc	193kB	19.41kB	1.54mJ	0.19mJ



Outline

- Motivation: challenges in the system-level design
- *Overview of methodology: concurrency mngnt*
- Cost-efficient run-time scheduling for RTOS
- Results on MPEG-4 IM1 player
- Long term research challenges

TCM steps aim at removing the bottlenecks for better performance



Optimized system specification

Inter-task DTSE

Task concurrency mngnt

Task conc. Extraction/trafo

1 2 Task/thread scheduling

2 Proc Array-processor allocation

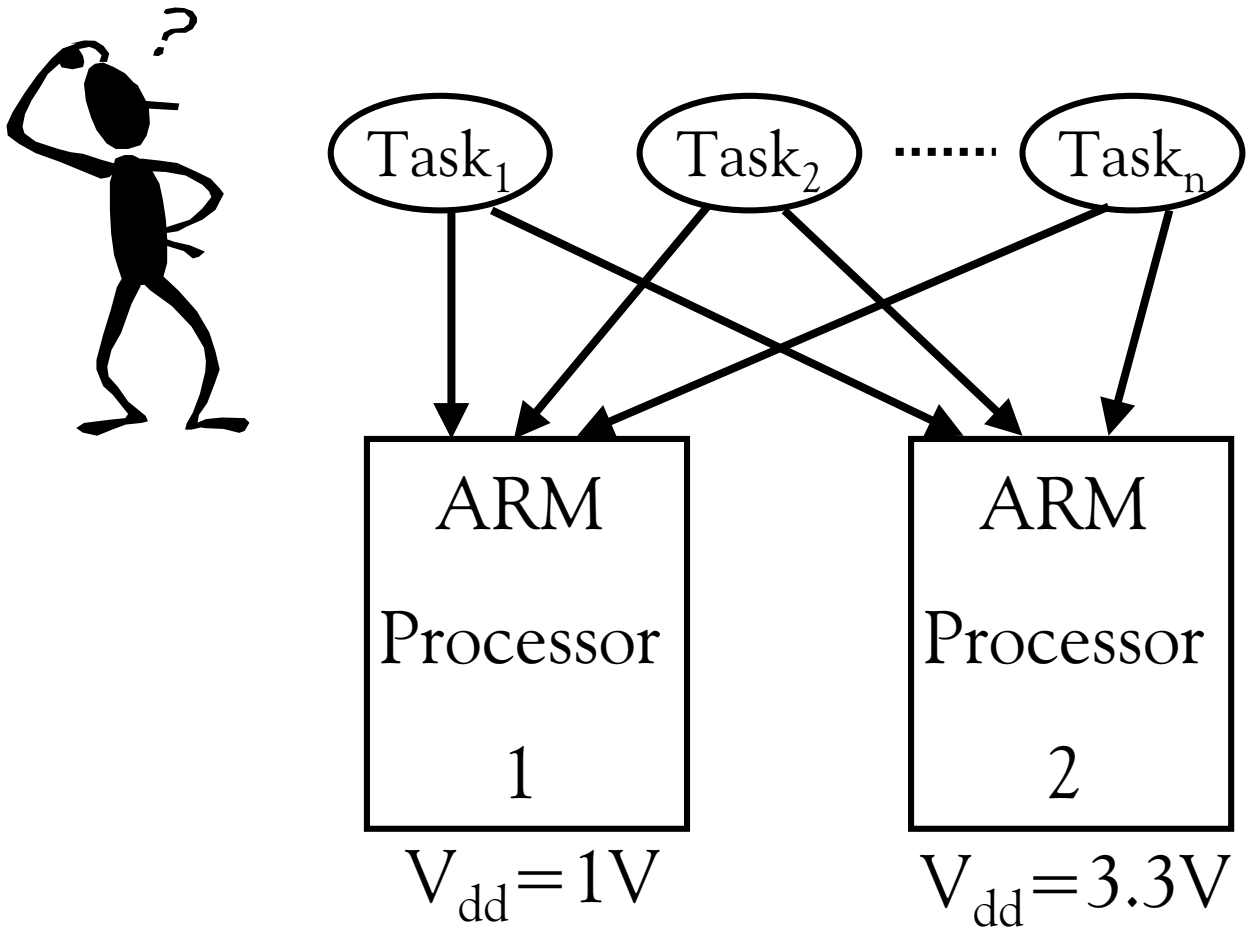
Task to processor assignment

Inter-task interface refinement

Task-level system architecture

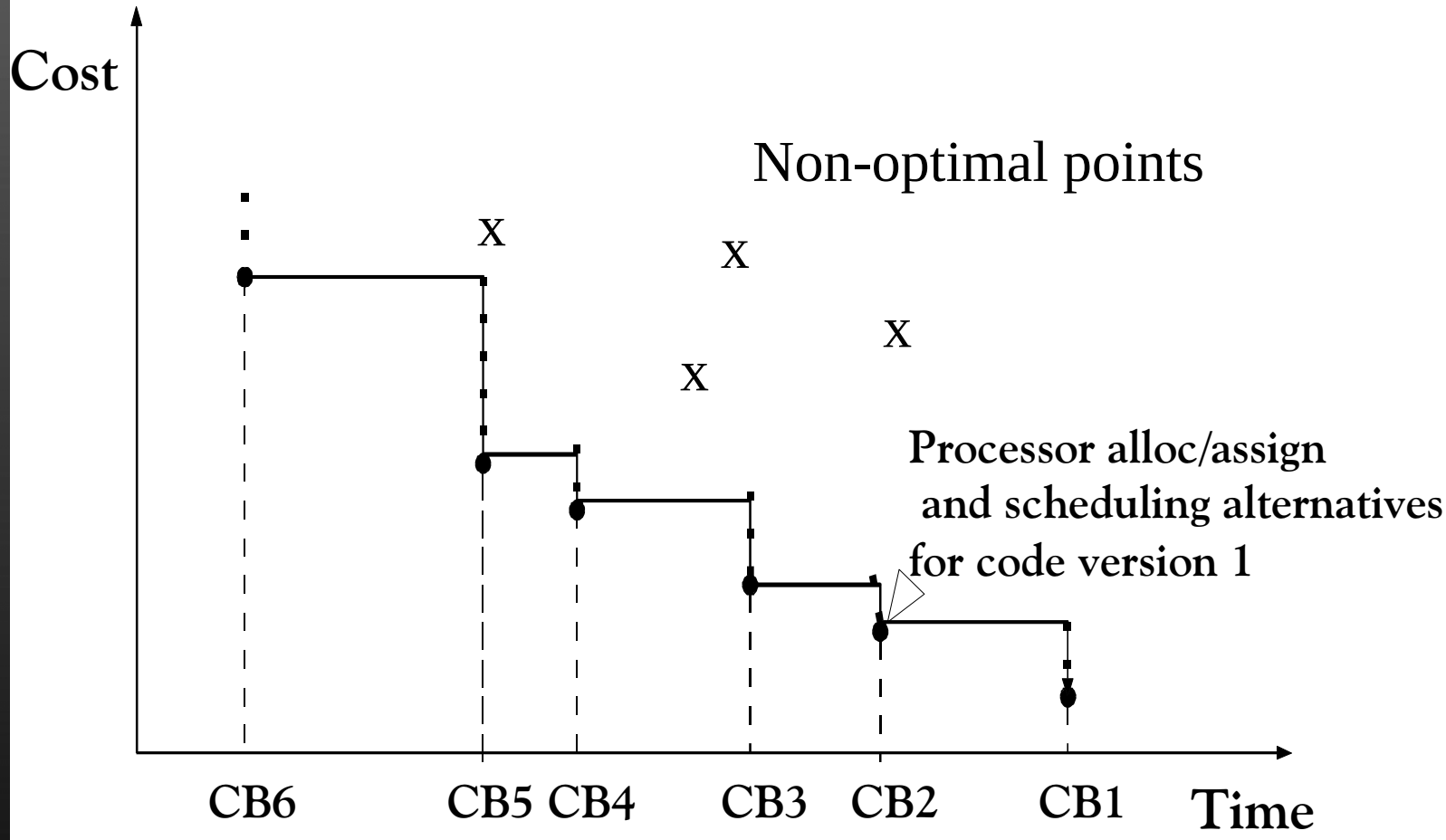
ICPP'00, Kluwer book'99

The 2-processor approach (scheduling + assignment)

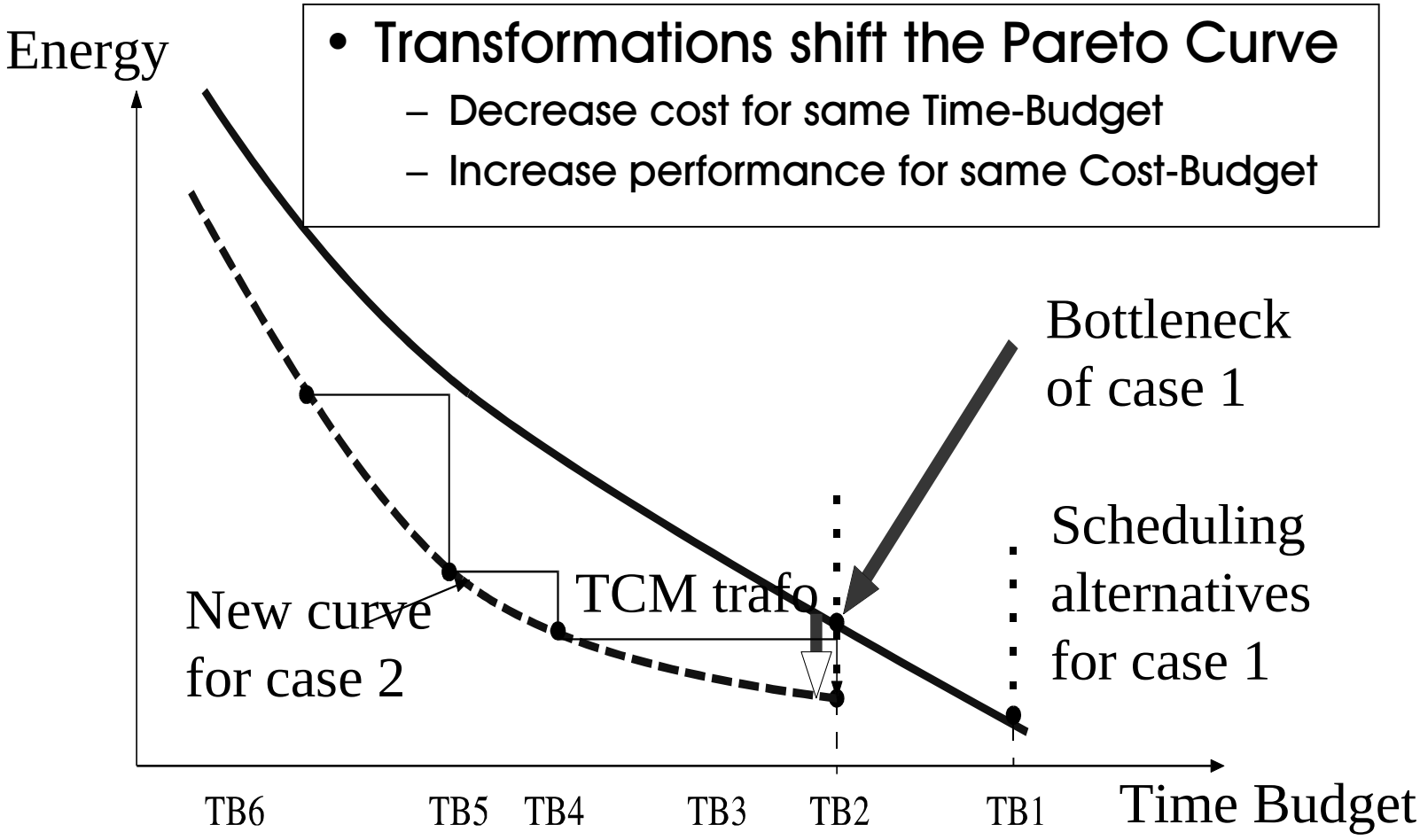


Codes'01, J.of Sys.Arch (summer 2001)

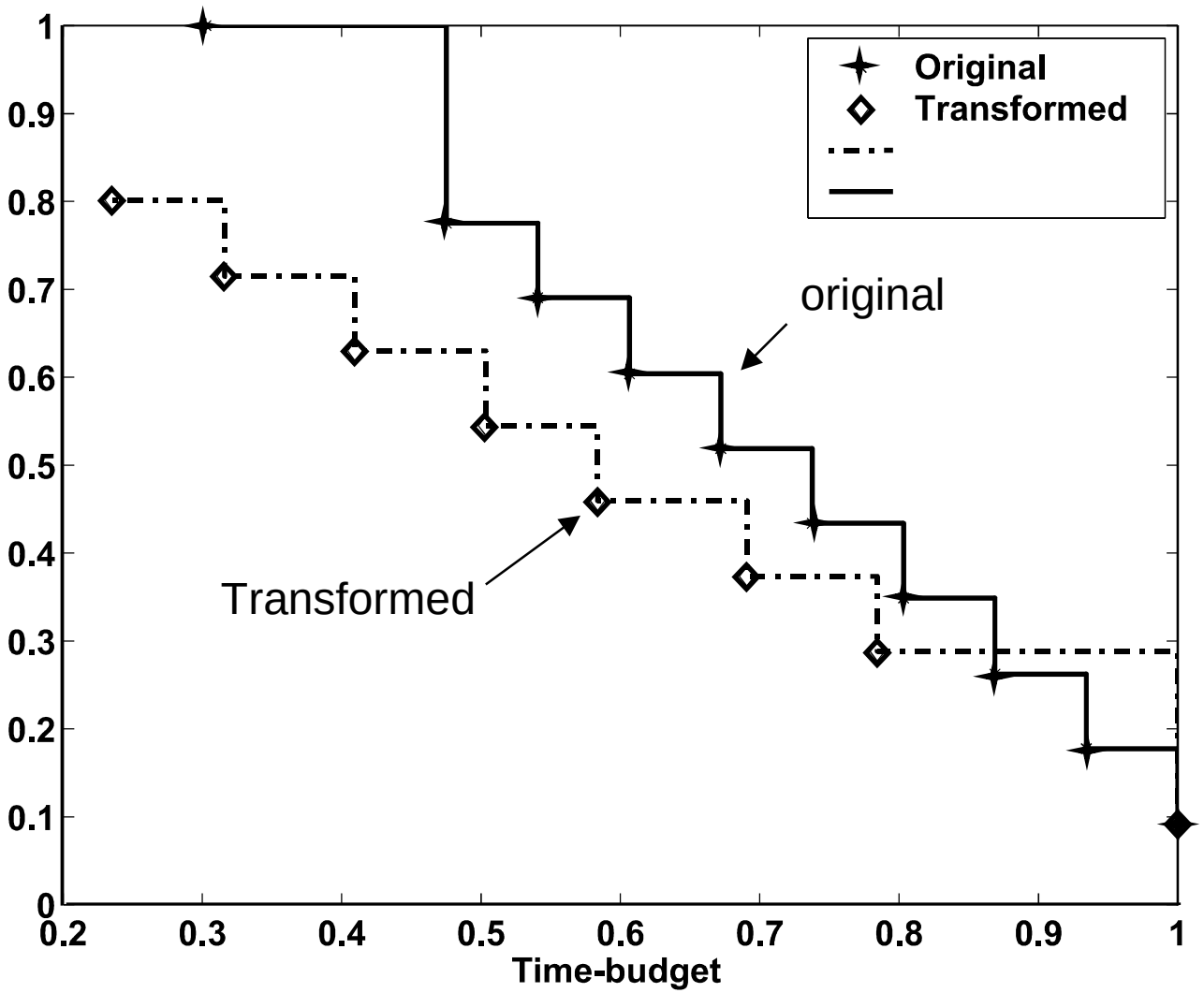
Trade-off between time budget (period/latency) and cost (e.g.energy) leads to Pareto curves



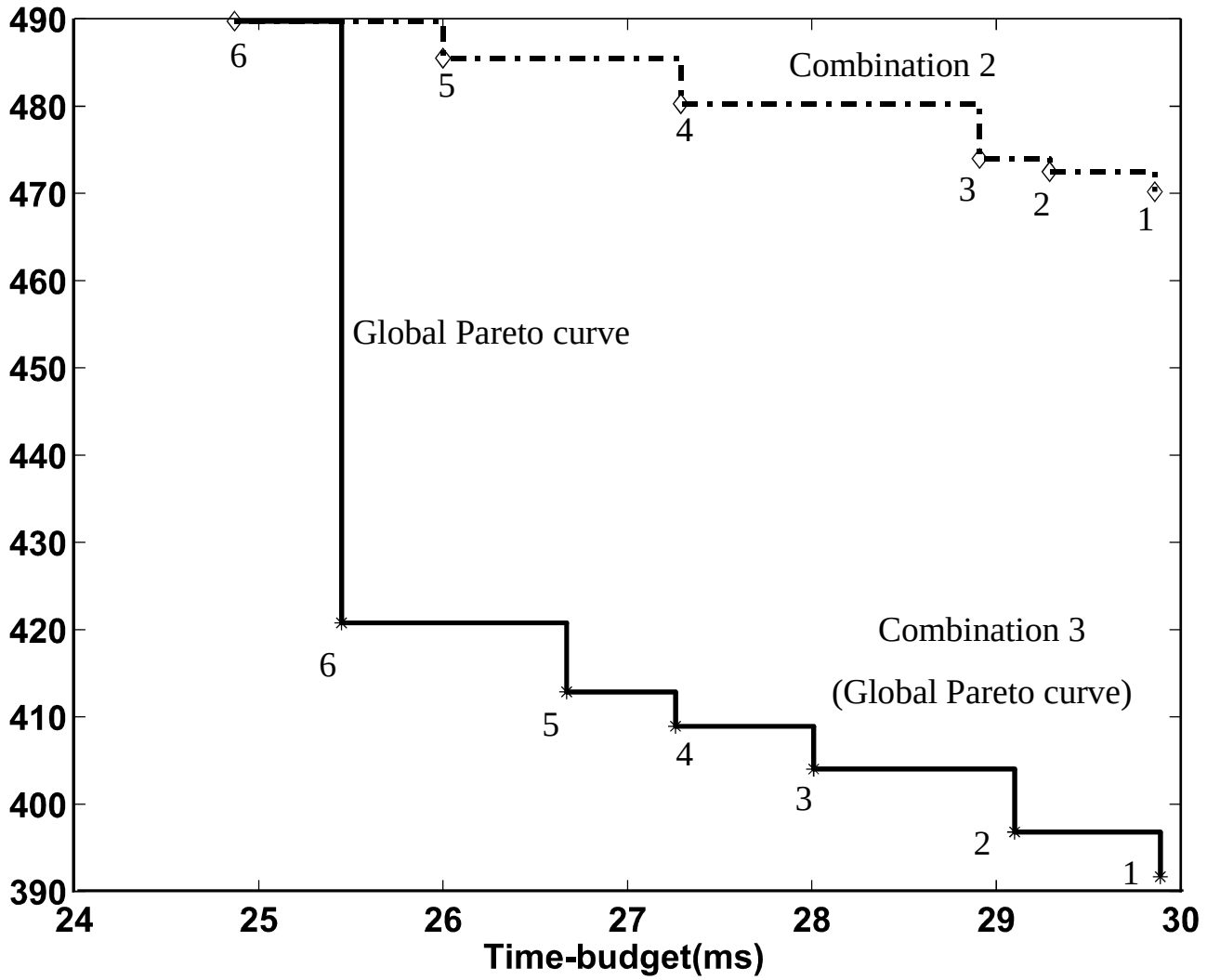
Pareto curve for 2 proc. mapping



Comparison for original and transformed graphs on 2 processors with different Vdd



Comparison between different processor platforms



Outline

- Motivation: challenges in the system-level design
- Overview of methodology
- *Cost-efficient run-time scheduling for RTOS*
- Results on MPEG-4 IM1 player
- Long term research challenges

Why run-time scheduler?(I)

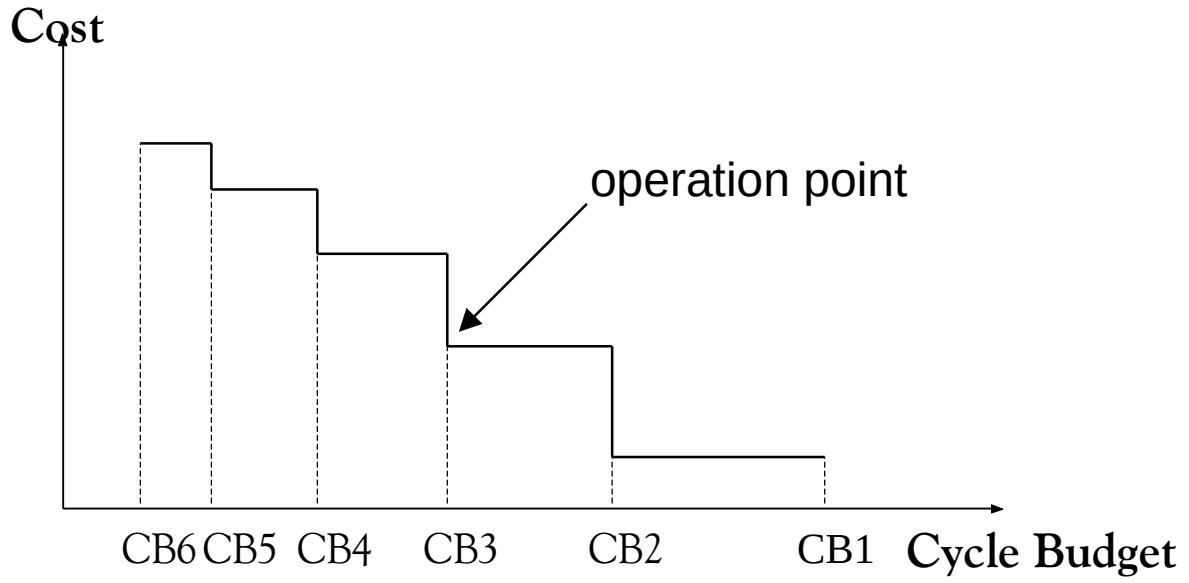
Design-time Scheduler

+ Predictability	- Schedulability for dynamic events
- Flexibility	
+ Run time complexity	+ Optimization

Run-time Scheduler

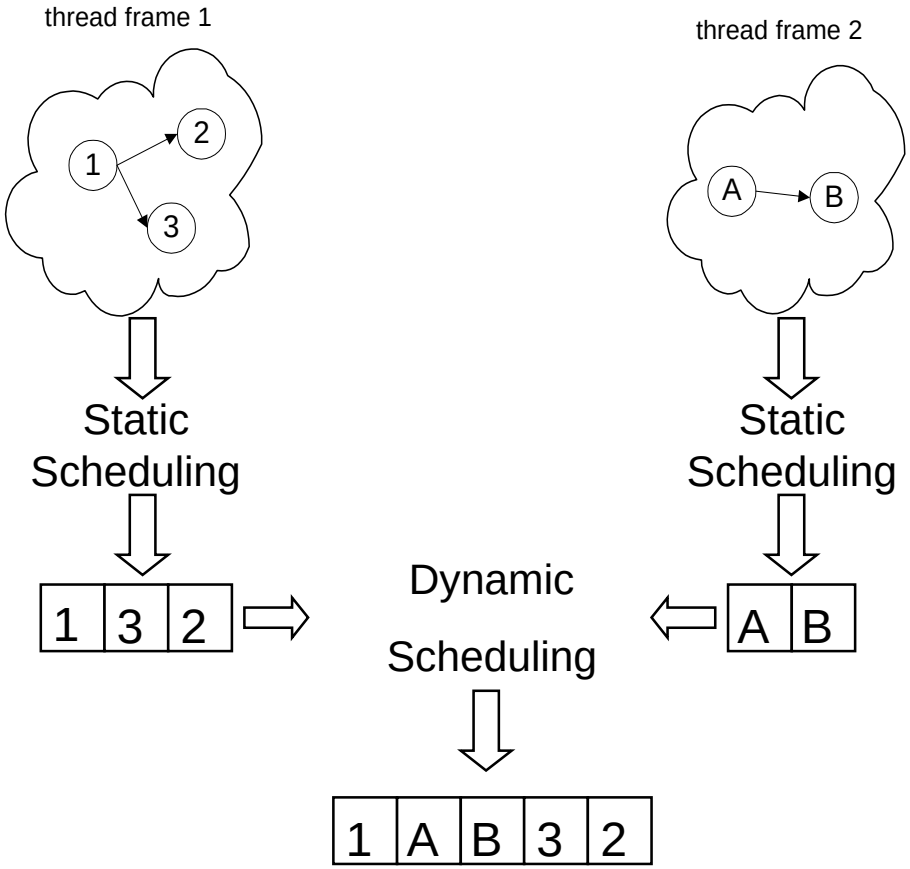
- Predictability	+ Schedulability for dynamic events
+ Flexibility	
- Run time complexity	- Optimization

Why run-time scheduler?(II)



- Design-time scheduler can only work at one operation point - no flexibility with changing environment
- It must consider the worst case - a waste when it's in other cases
  - data dependency
  - non-determinism
  - dynamic creation/release

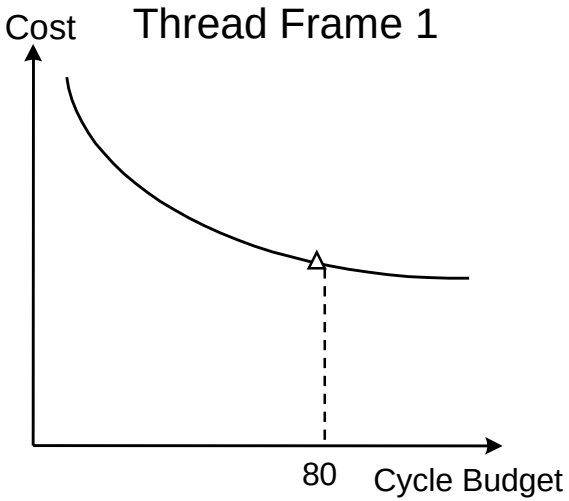
Combination of design- and run-time schedulers

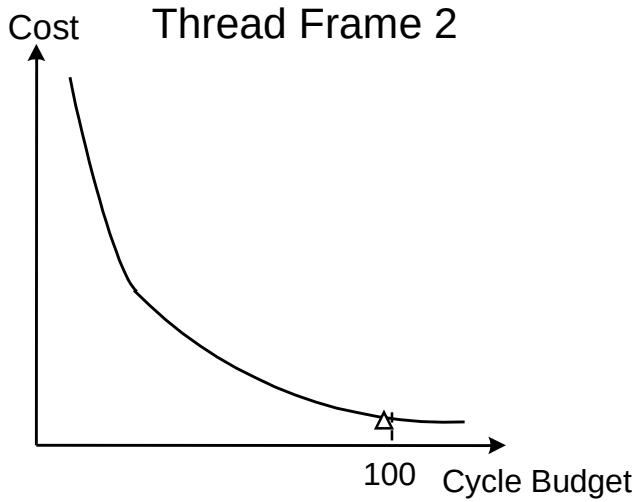


- Design-time scheduling: at compile time, exploring all the optimization possibility
- Run-time scheduling: at run time, providing flexibility and dynamic control at low cost

Cases'00, Design&Test- Sep.'01

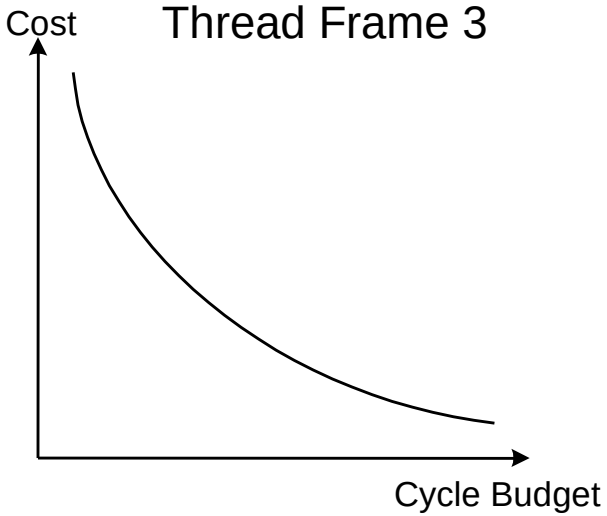
Run-time scheduling example(I)



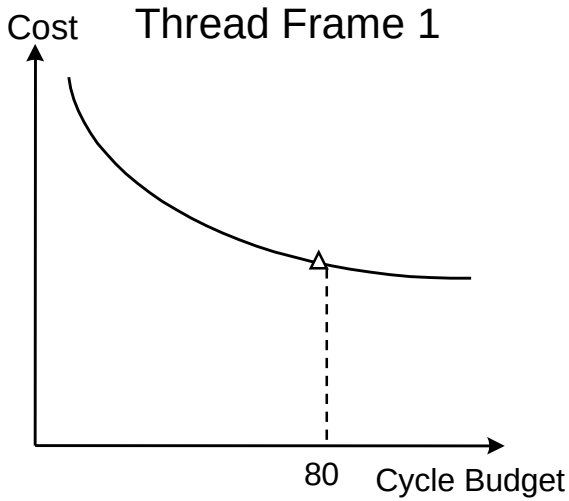


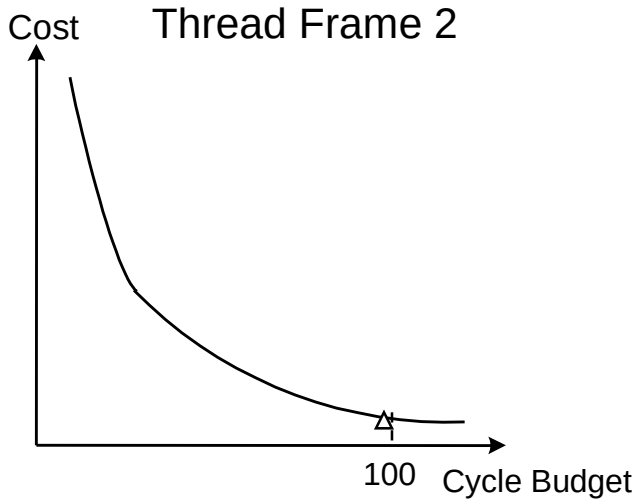
\+

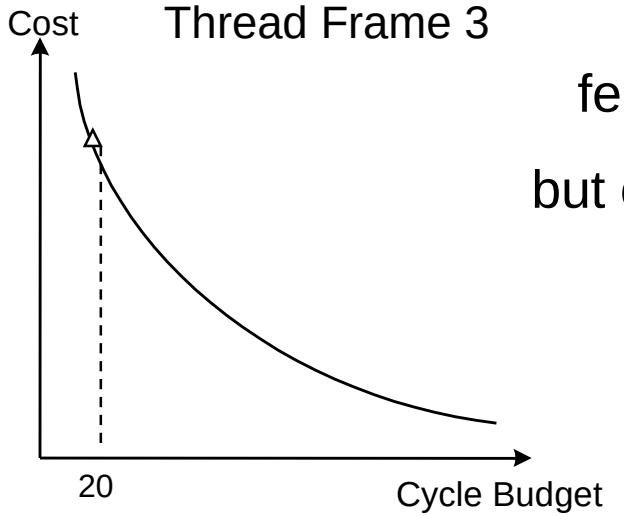
- A new thread frame coming
- 20 cycle budgets available



Run-time scheduling example(II)



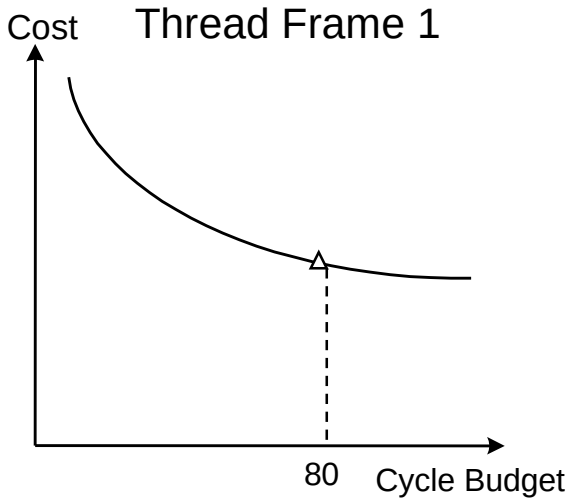


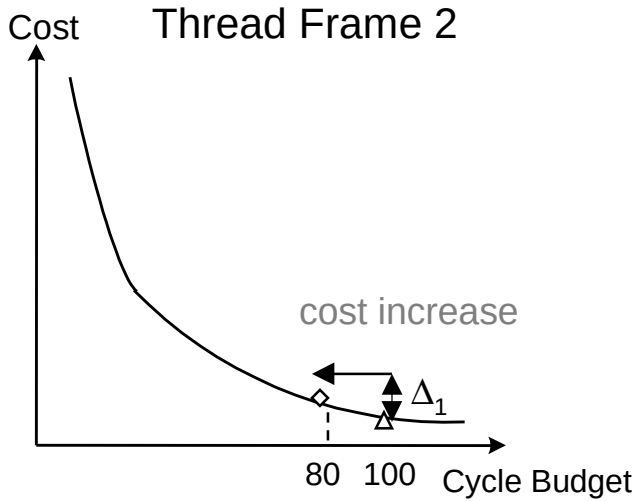


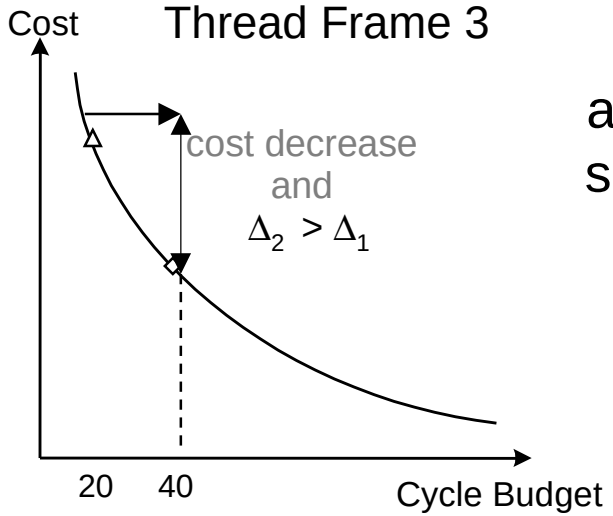
feasible,

but optimal?

Run-time scheduling example(III)

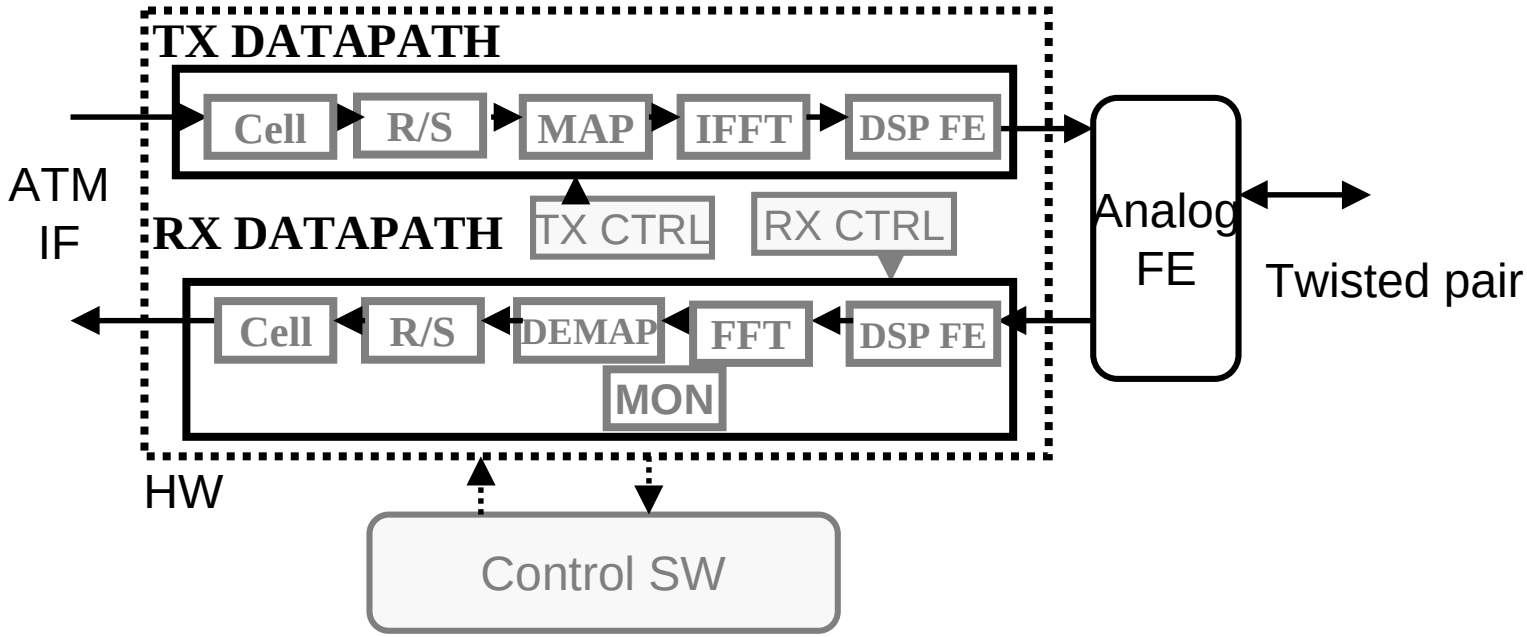






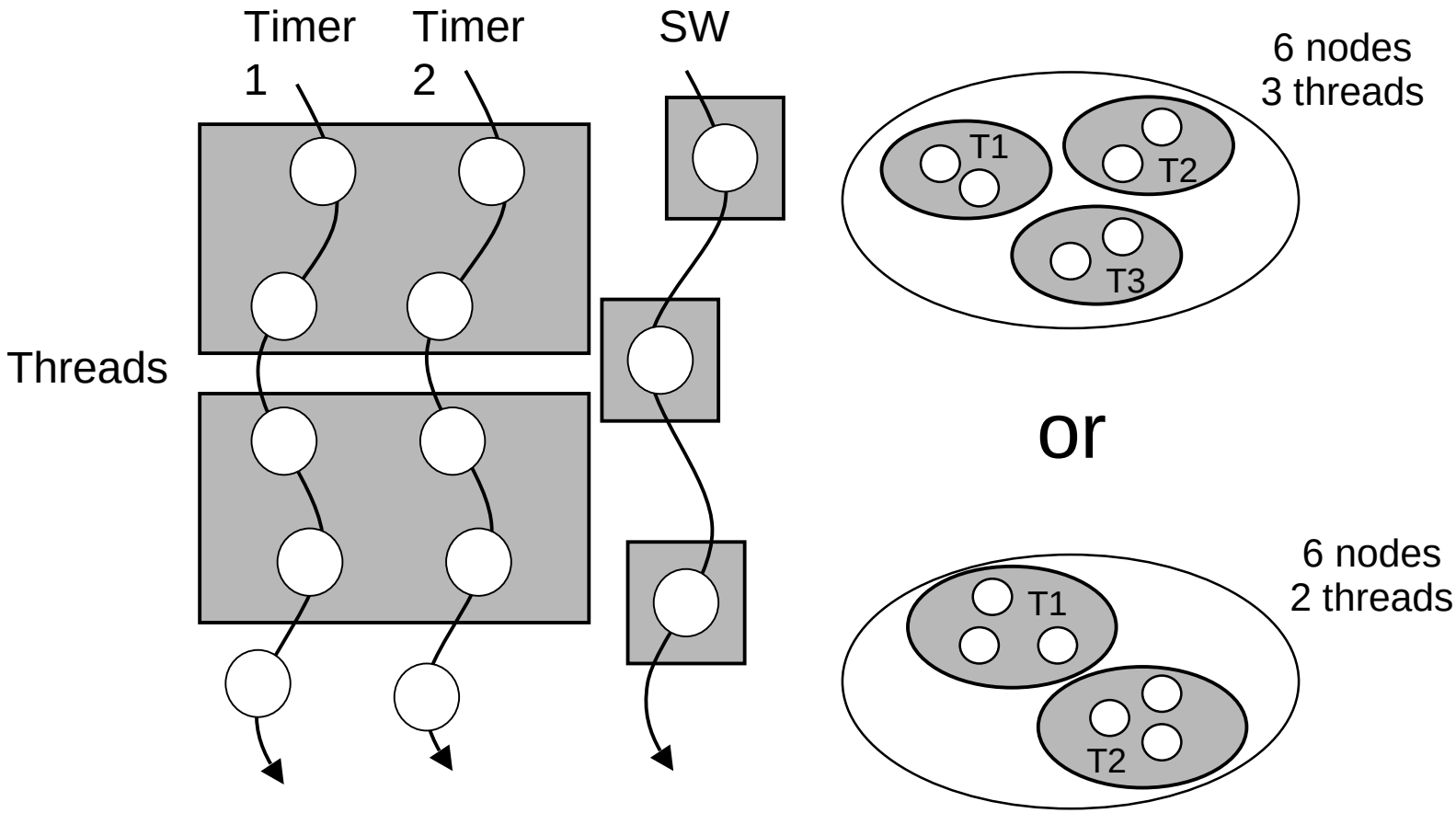
a better solution

ADSL System Architecture

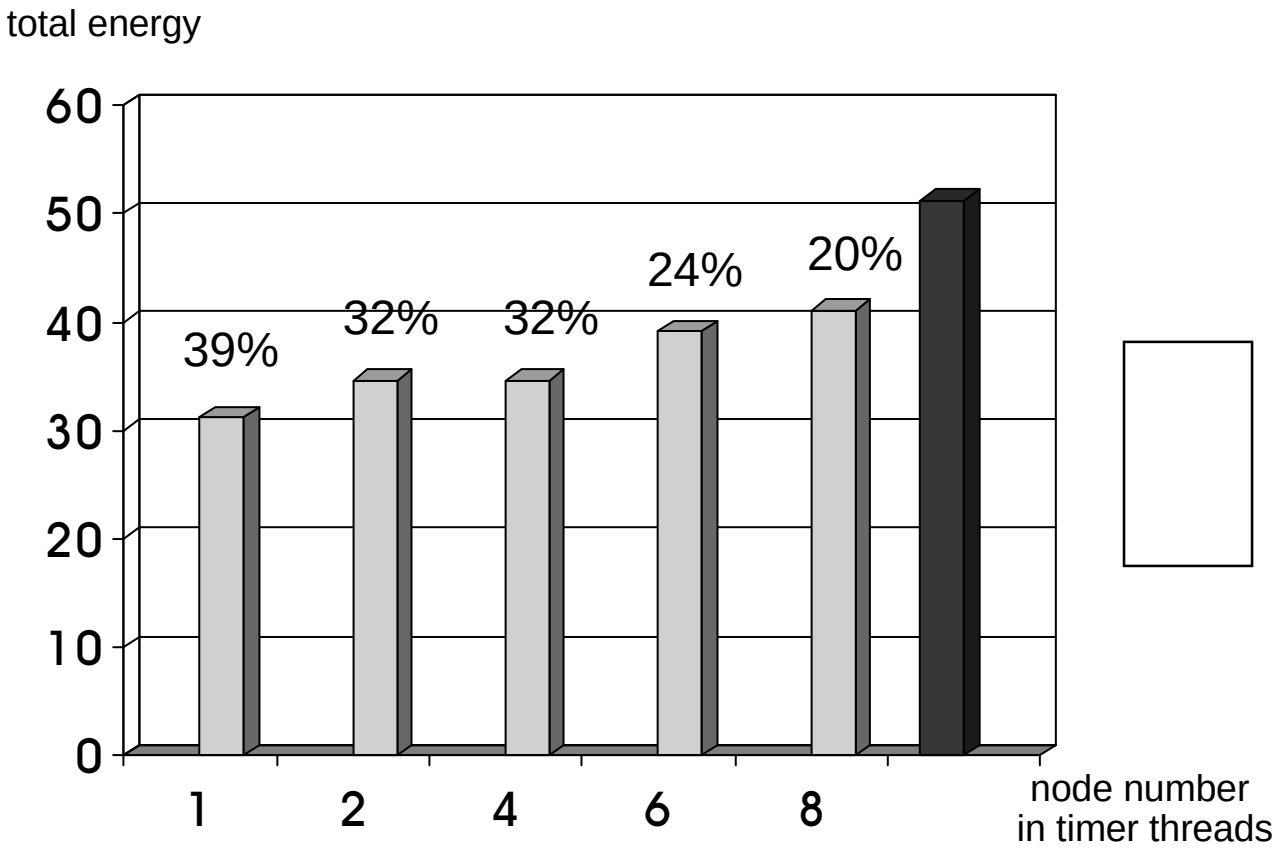


- TX and RX CTRL have a deadline equal to one symbol, where f_{symbol} = 4 kHz
- SW tasks have deadlines much longer, for example, 128 symbols

Granularity of the threads can be important



Run-time scheduling result



Cases'00

Main messages

- Embedded multi-media applications are becoming very dynamic and concurrent in nature
  => RTOS essential
- Task Concurrency Management approach provides the flexibility and optimization possibility while limiting the run time computation complexity
- A multiprocessor platform with different working voltages potentially provides an energy saving solution
- Application-specific run-time scheduling technique combined with design-time scheduling to provide cost-performance Pareto-curve essential for effective solution

Outline

- Motivation: challenges in the system-level design
- Overview of methodology
- Cost-efficient run-time scheduling for RTOS
- Results on MPEG-4 IM1 player
- *Long term research challenges*

Research challenges

- Extract “grey box model” from conventional specifications
- Classification and design support for grey box transformations
- Find fast heuristics for design-time and run-time scheduling methods
- Consider the communication and context switch overhead
- Fully handle complex non-deterministic behaviors