# Proactive Fine-grain Sub-task Scheduling in Real-time Embedded Systems

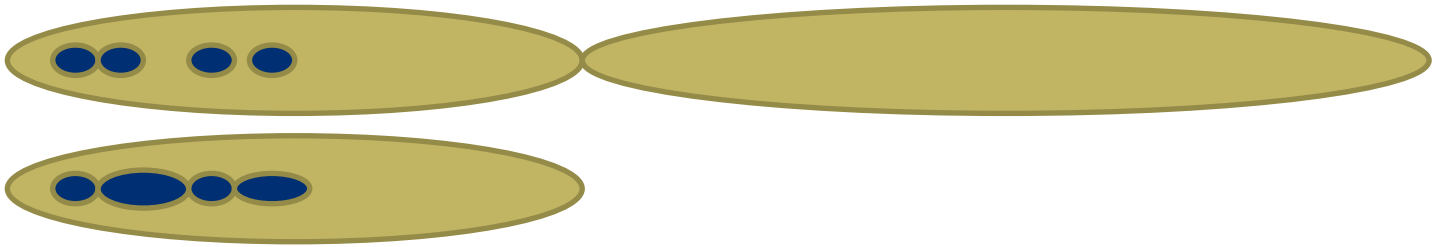
Satya Munaga

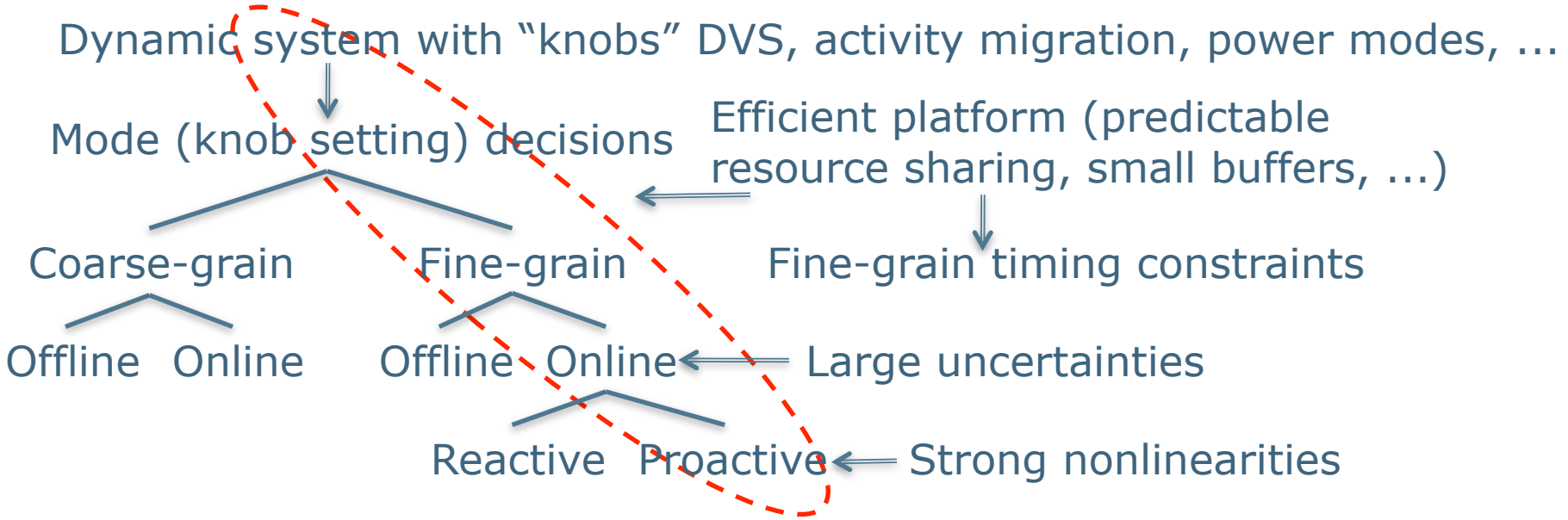
Promoter: Francky Catthoor

# Outline

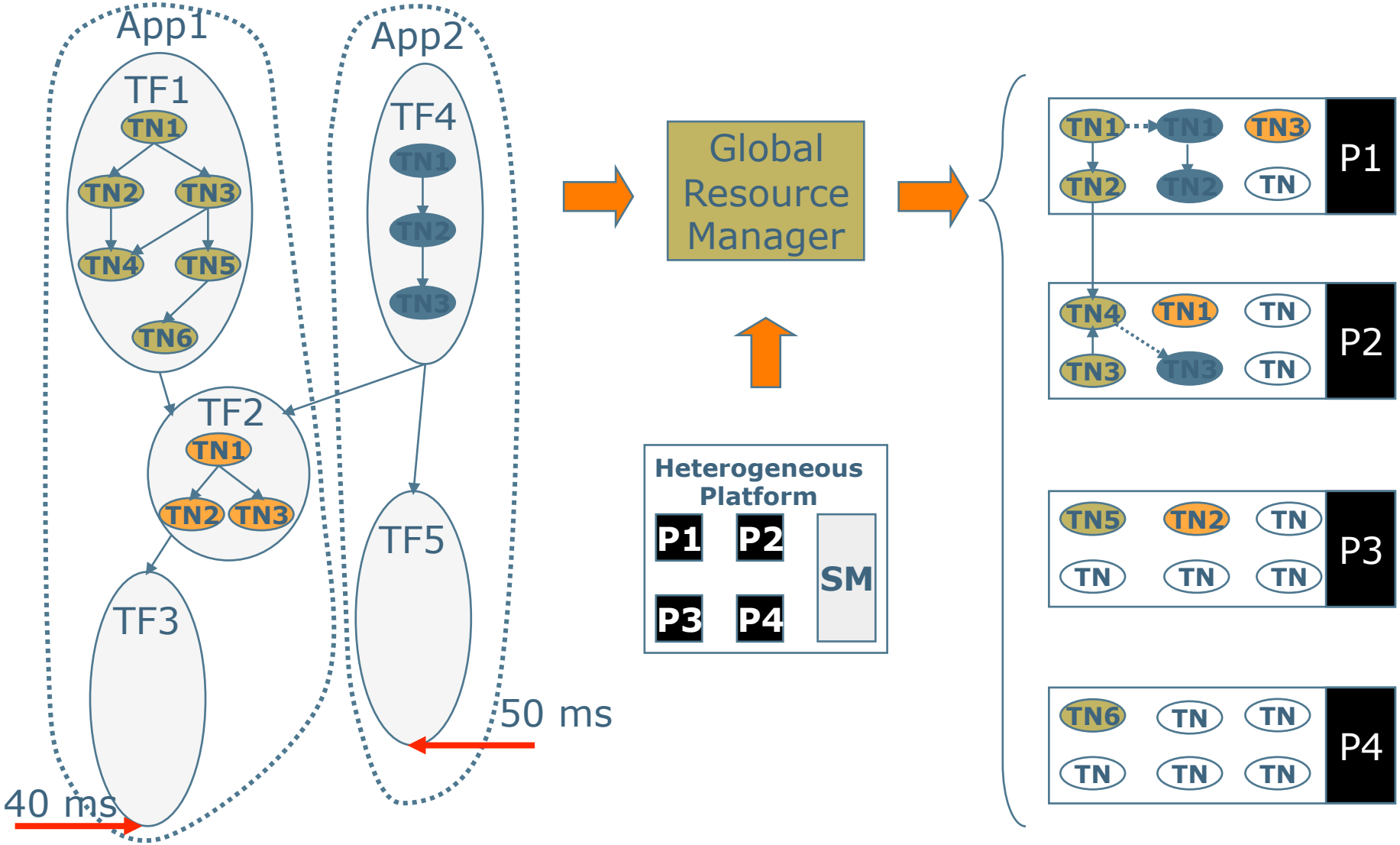
- Online fine-grain sub-task scheduling
- State-of-the-art
- Nonlinearities
- Proactive control
  - Upper bound refinement
  - Likely-case optimization
- Experimental results
- Contextual turbo-modes

# Proactive Fine-grain Sub-task Scheduling



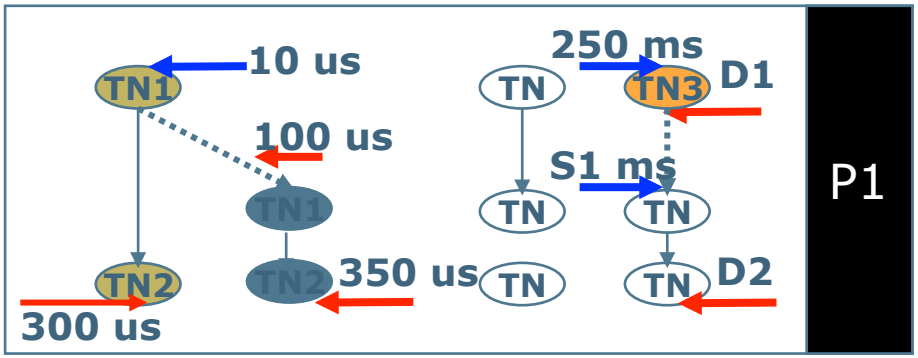


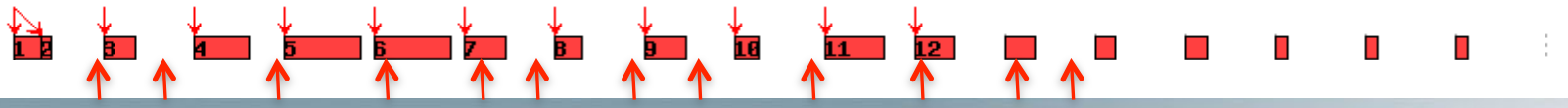
# Hierarchical Resource Management



# Local Resource Management Problem

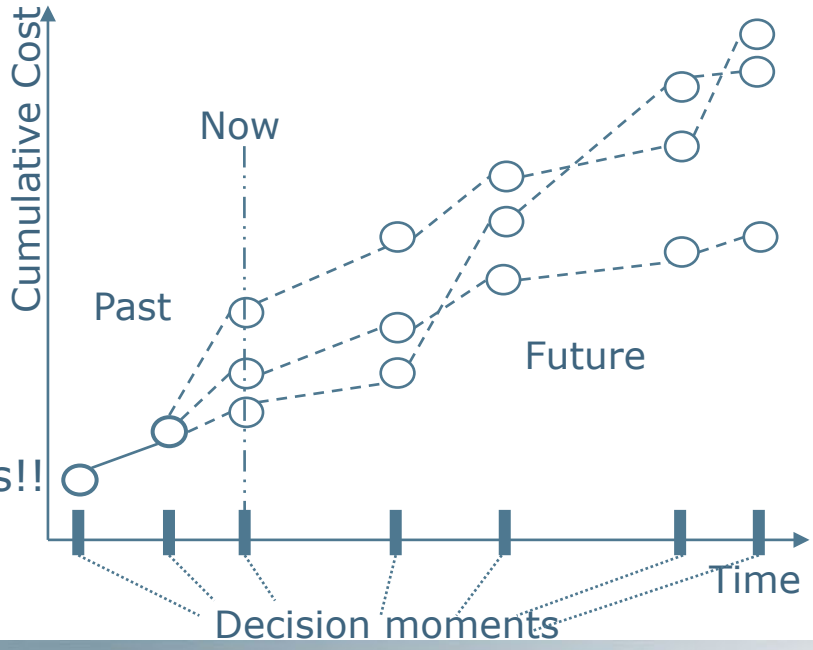
- Fine-grain timing constraints
- Further restrict freedom to reduce overhead
  - E.g., don't change the ordering, end times as deadlines
- Online LRM problem specification
  - Stream of fine-grain sub-tasks; release time and deadline for every sub-task; post deadline mode to ensure schedulability;
  - decide the execution mode of a sub-task just-in-time; minimize energy consumption; **no deadline miss**!



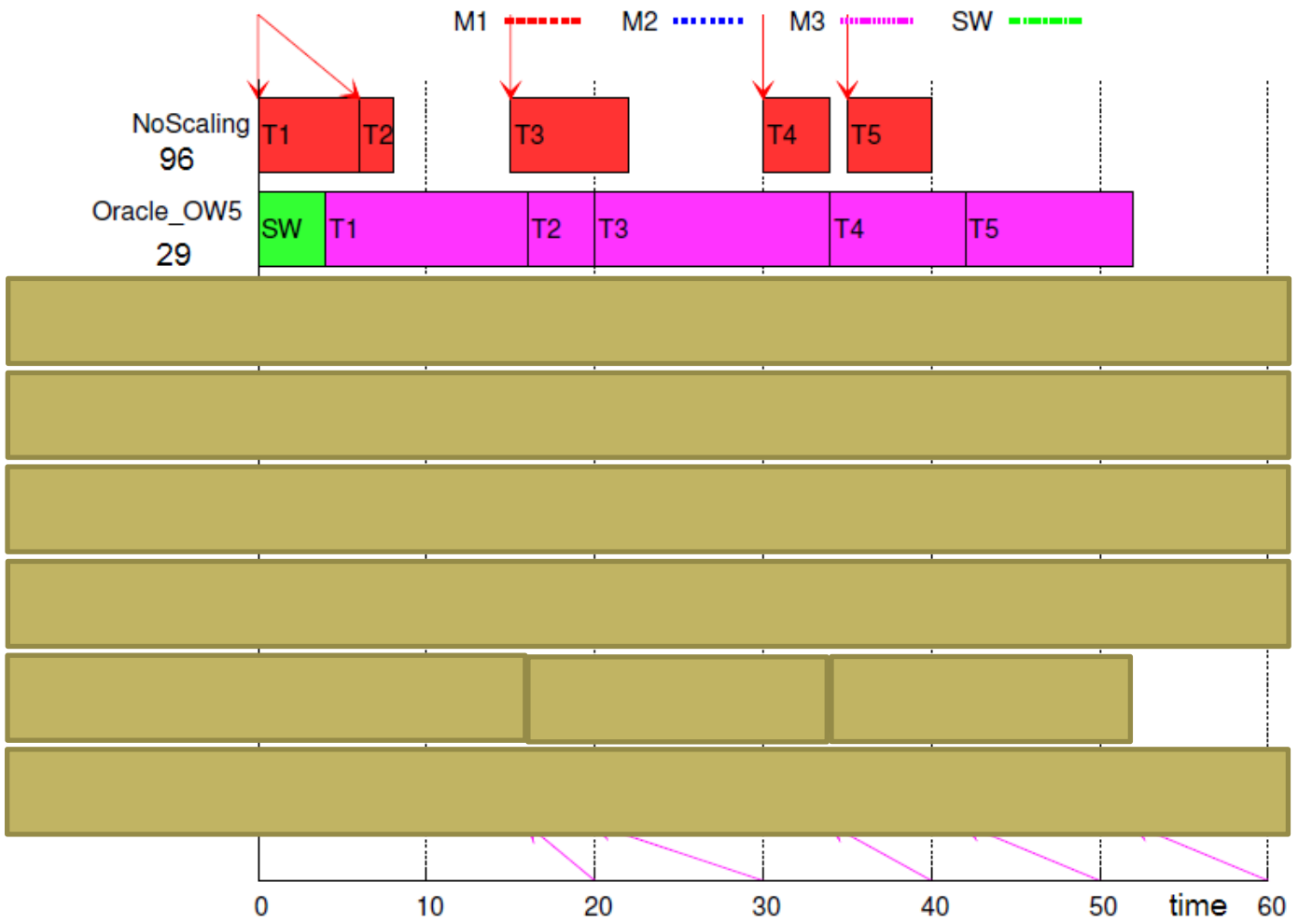



# Uncertain + Nonlinear => Time-linkage

- Uncertainties
  - Unknown at the time of decision making
  - But influence constraints (deadlines) and costs (energy)
  - E.g., execution-time or workload
- Nonlinearities
  - Non-negligible switching overhead
  - Discrete modes
  - Release times
- Time-linkage
  - Strong state influence!
  - Knowing the future in advance helps!!



# Task Scheduling Example

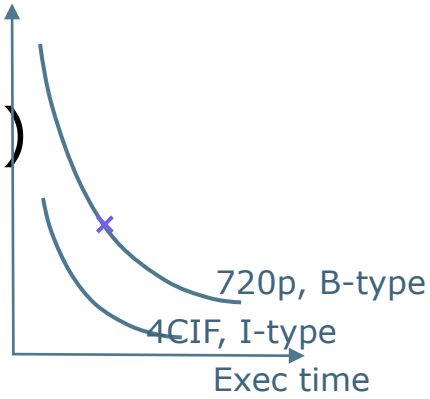


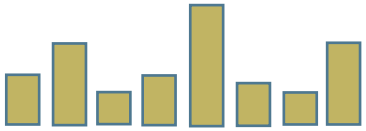
# Proactive Control

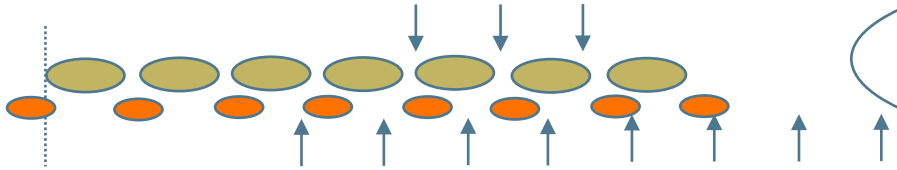
- N.B.: uncertainties @ decision-making time
- Solve constrained optimization online
  - For present + (part of) future together
    - Revisit the decisions as time progresses, if necessary
    - Reuse of concepts from model predictive control
  - RT refined upper bounds (vs. design-time upper bounds)
    - Reuse system scenarios for sub-problems
  - Formulate the optimization problem for the expected case
    - In contrast to worst-/average-/typical-/scenario-case
    - Use of predictors/estimators (NOT 100% guaranteed)
    - Ensure feasibility (upon re-decision) for any viable-case
  - Dynamic procedure to further reduce decision overhead

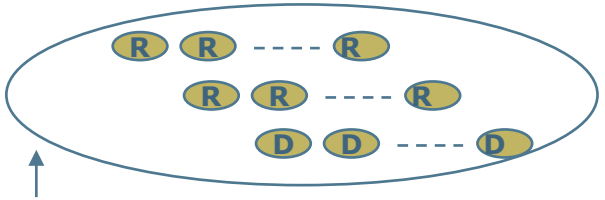
# A Video Case Study

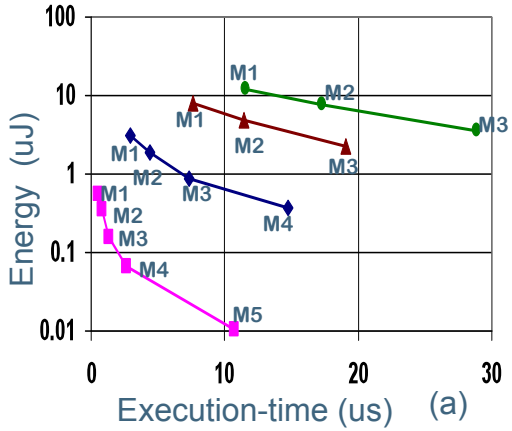
- Video decoding TF (~30ms per frame)
- TF-level scenarios (fmt., res., fr-typ, …)
- MB-level data pipelining (~11us)
- DVFS-like knob: 5-20us SWO

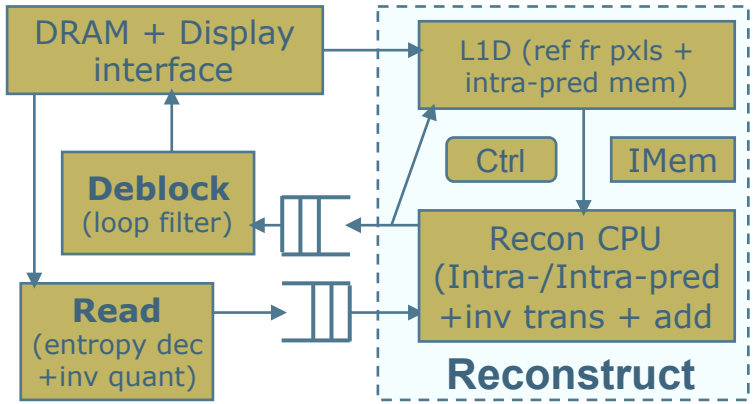


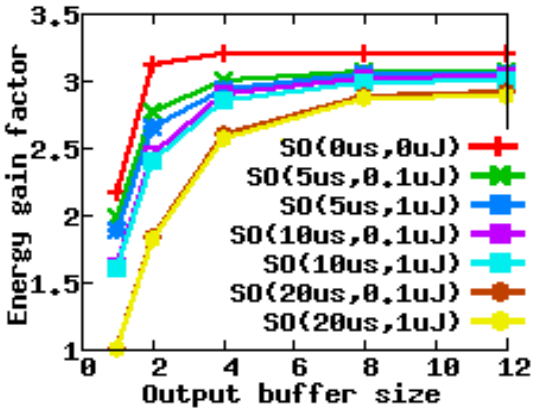







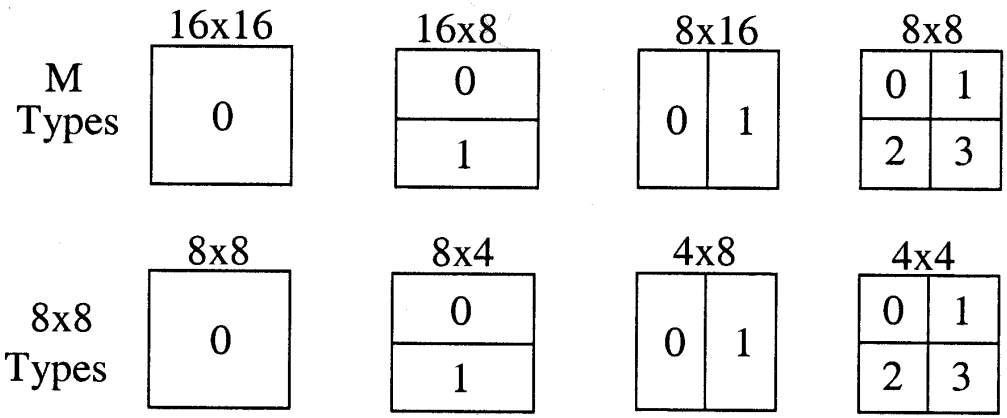



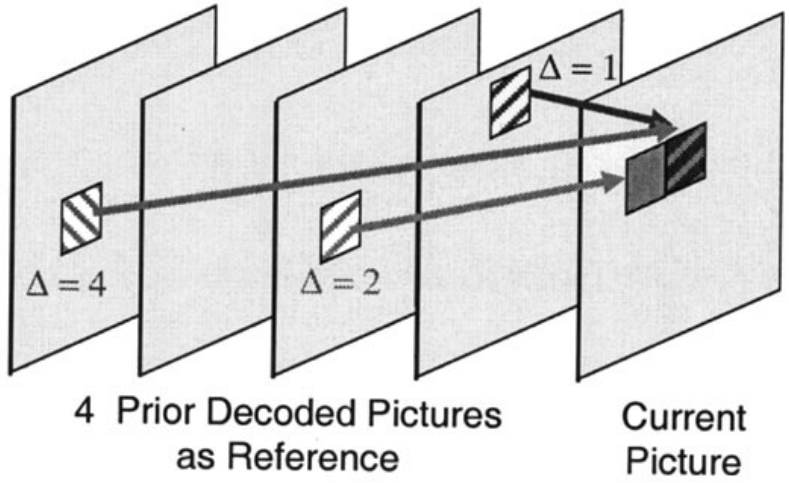



# MB-level System Scenarios for UBR

## Exploit correlations




## Cluster frequently occurring cases

SC1
164

Intra Type MBs

SC2
1292

P16x16
1 MV

SC3
2660

P16x16
2 MV

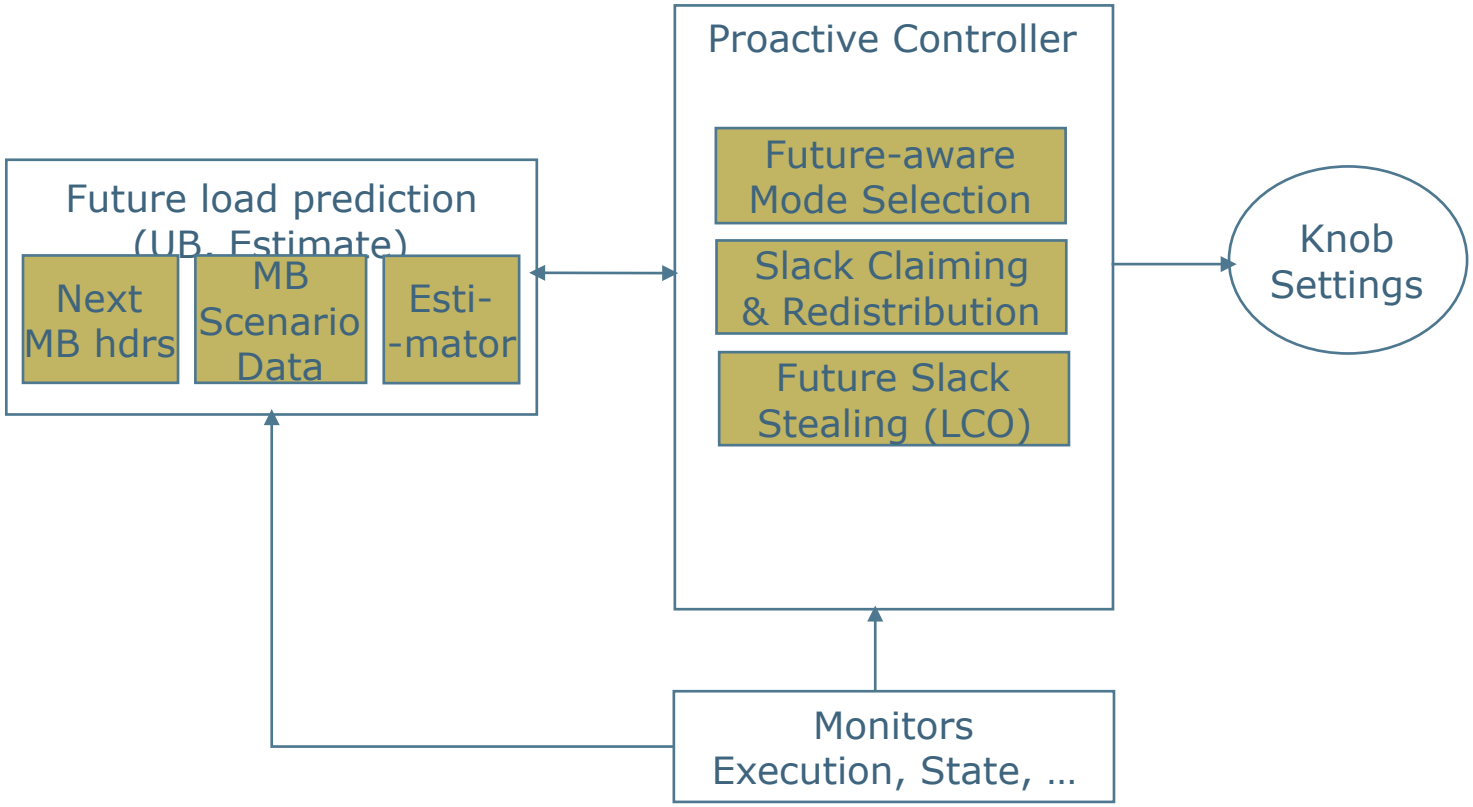
Backup Scenario
5787

Rest

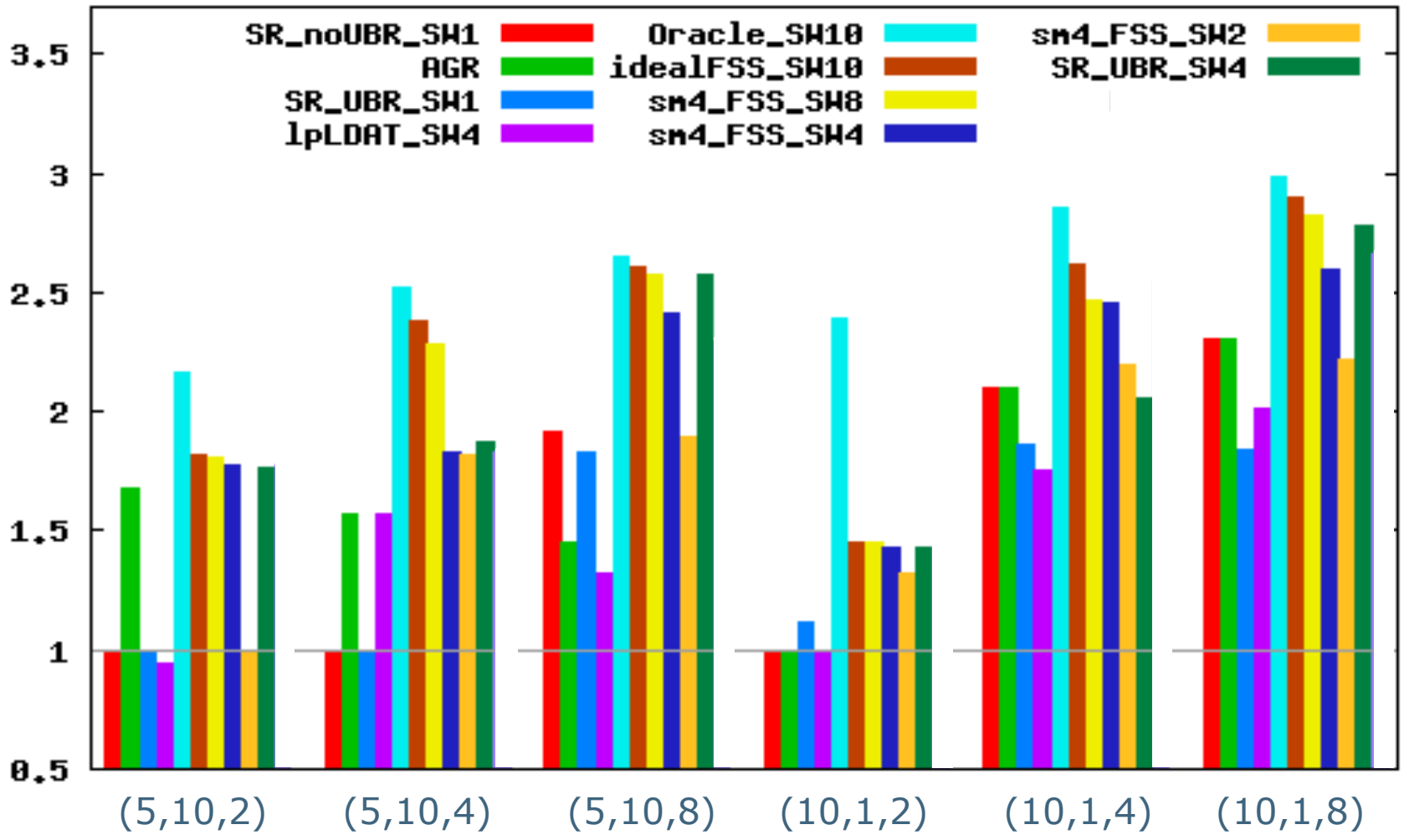
<20%

Tighter guaranteed upper bounds on cycles

# Proactive Controller for Performance-Scaling



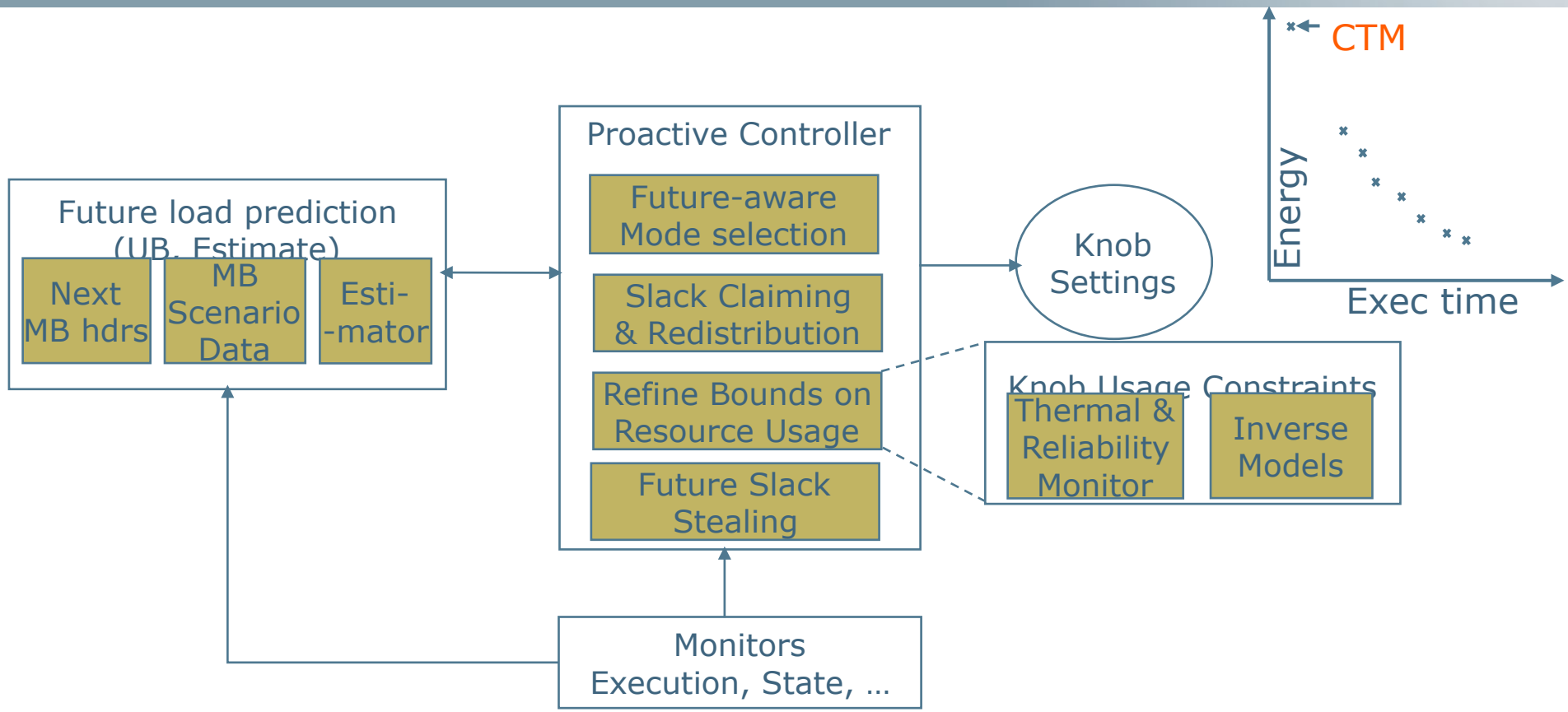
# Results


SR_noUBR_SW1
AGR
SR_UBR_SW1
lpLDAT_SW4
Oracle_SW10
idealFSS_SW10
sm4_FSS_SW8
sm4_FSS_SW4
sm4_FSS_SW2
SR_UBR_SW4
3.5
3
2.5
2
1.5
1
0.5
(5,10,2)
(5,10,4)
(5,10,8)
(10,1,2)
(10,1,4)
(10,1,8)

# Contextual Turbo-modes (Gas-pedal)

- Pareto-optimal operating points (i.e., better in at least one aspect) but not always available
- Type1: better in all aspects
  - Always good to use if available
- Type2: worse only in cost aspects
  - Using when available will not interfere with constraints
- Type3: worse only in constraint aspects
- Type4: worse in both cost & constraint aspects
- Type3 & Type4
  - Long usage leads to violation of some constraints
  - Normally non-existent in traditional designs

# CTM/Gas-pedal => Closer to "Crystal Ball"!



# Conclusions

- Very fine-grain sub-task scheduling with hard timing constraints
  - Efficient & predictable platforms
  - Uncertainties + nonlinearities => time-linkage
- Proactive control
  - Look-ahead
  - Dynamic bounding (upper bound refinement)
  - Likely-case optimization
  - Repeat as execution progresses (as and when needed)
- Contextual turbo-modes
  - Modes that live only short time!
  - Useful and should be designed-in hard real-time systems!!


25
imec
Years of Making
Technology Fly