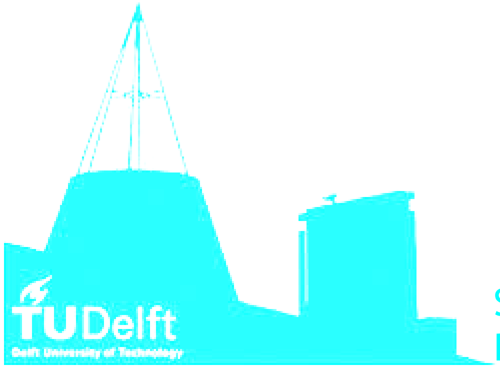
ETH Zürich

Pengcheng Huang
Georgia Giannopoulou
Lothar Thiele

Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

RTAS 2016



Sujay Narayana
R.Venkatesha Prasad

Why?

Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

Why?

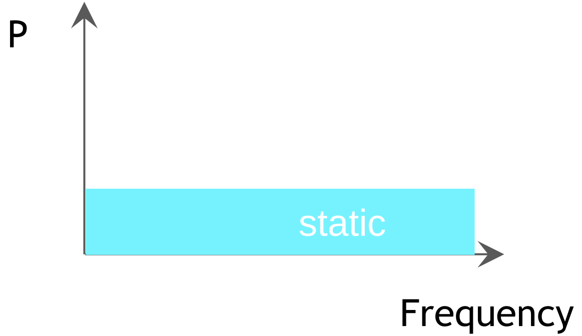
Embedded system industry is moving towards MC, but little has been done on

"Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores"

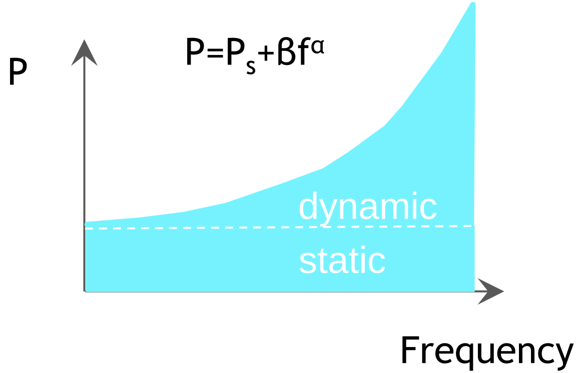
Exploring Energy Saving for
Mixed-Criticality Systems on Multi-cores

Exploring Energy Saving for

Mixed-Criticality Systems on Multi-cores

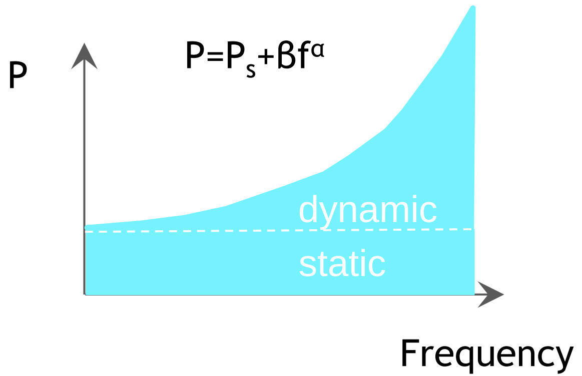


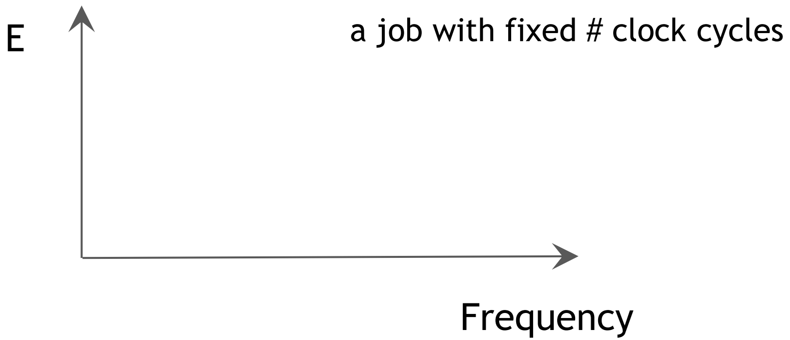
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores



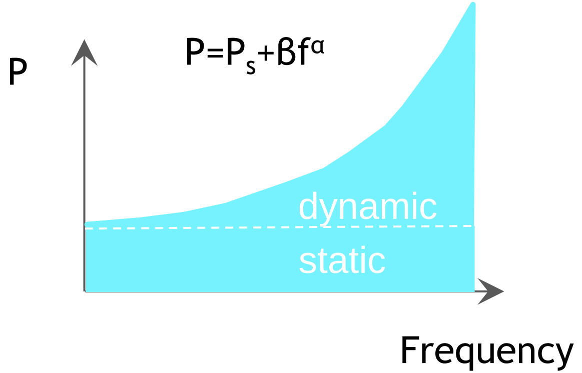
Exploring Energy Saving for

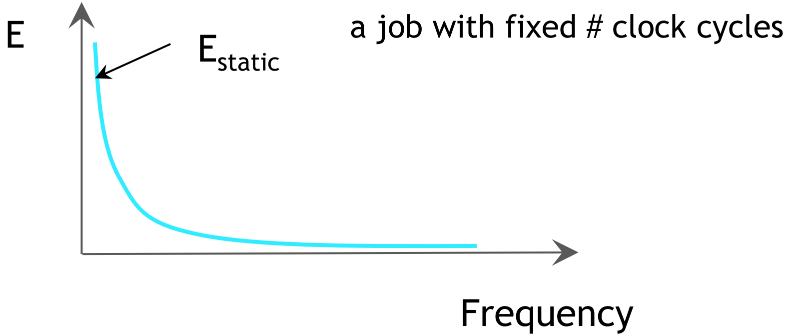
Mixed-Criticality Systems on Multi-cores



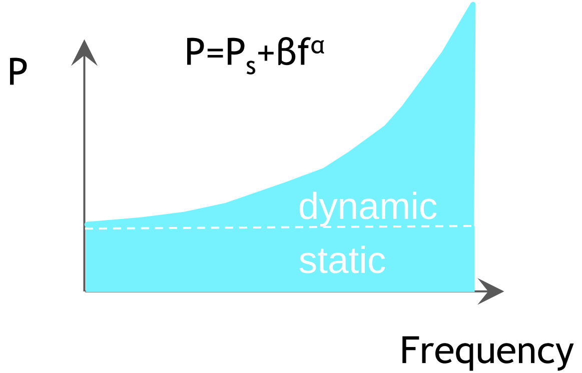


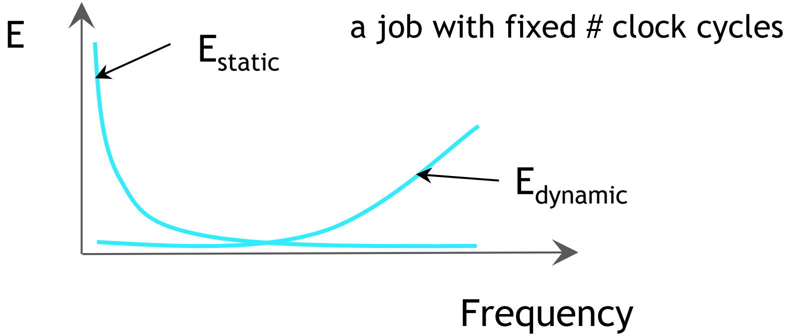
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores



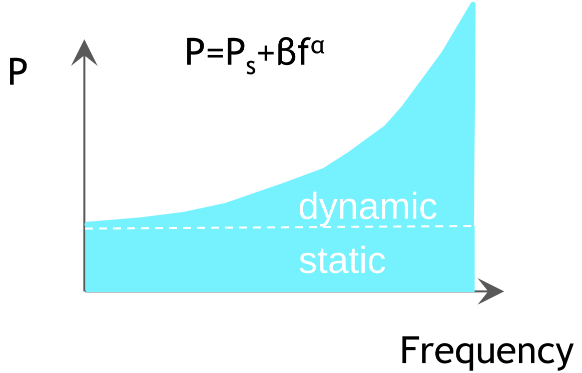


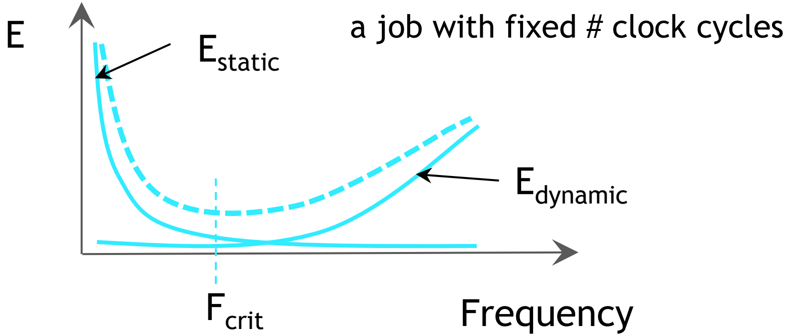
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores



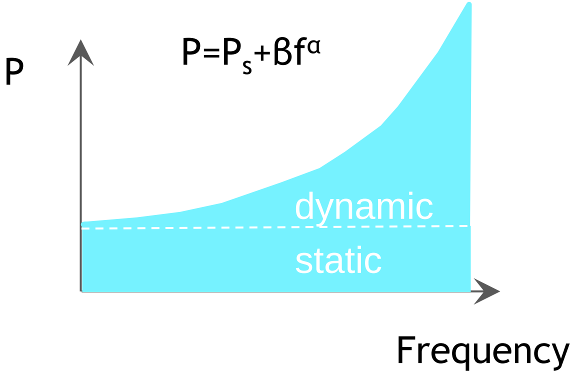


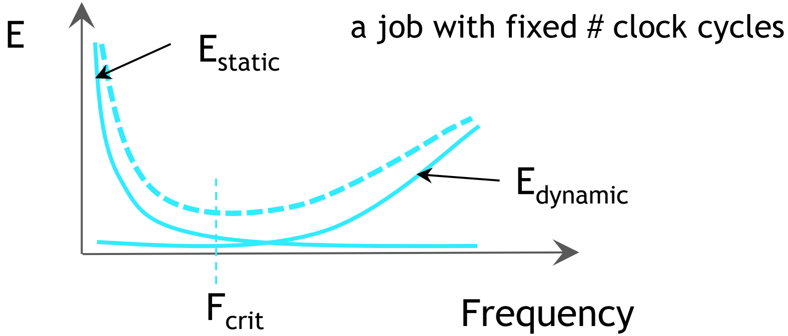
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores



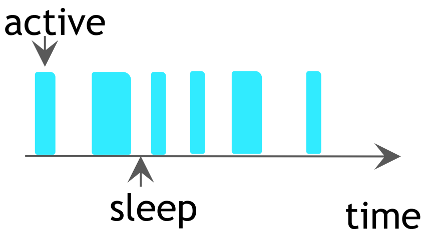


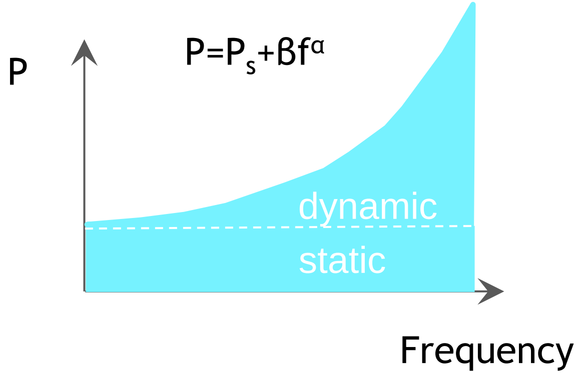
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

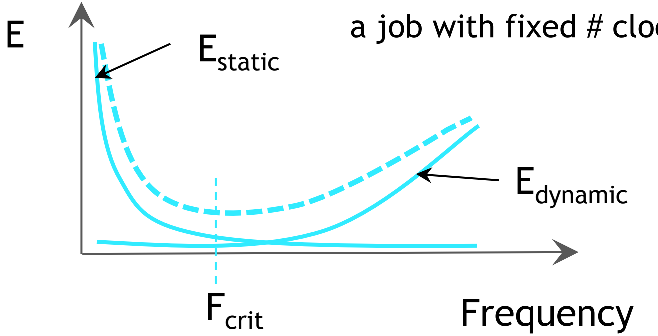




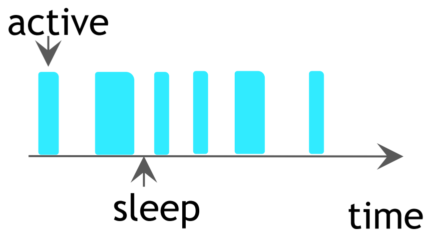
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores







Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores



DVFS, frequency >= F_{crit}

no energy dissipation when sleep

zero overhead sleep <-> active

find frequencies → minimize energy
well studied problem for classical systems

Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

apps of varying criticality levels
share a common platform
time overrun is a concern

asymmetric isolation

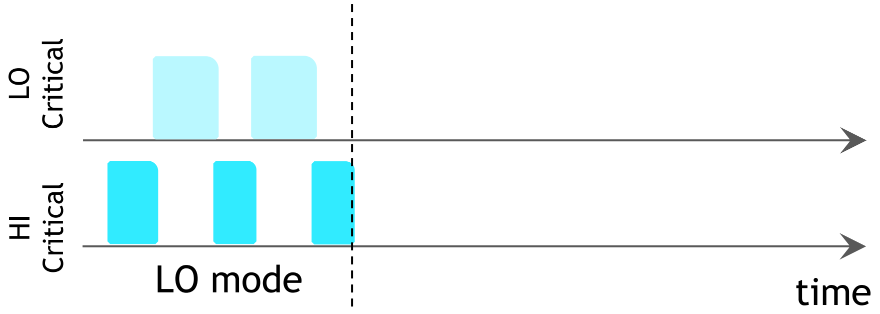
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

apps of varying criticality levels

share a common platform

time overrun is a concern

asymmetric isolation



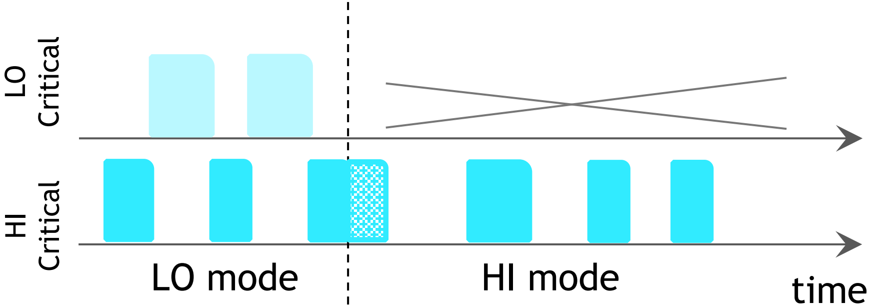
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

apps of varying criticality levels

share a common platform

time overrun is a concern

asymmetric isolation



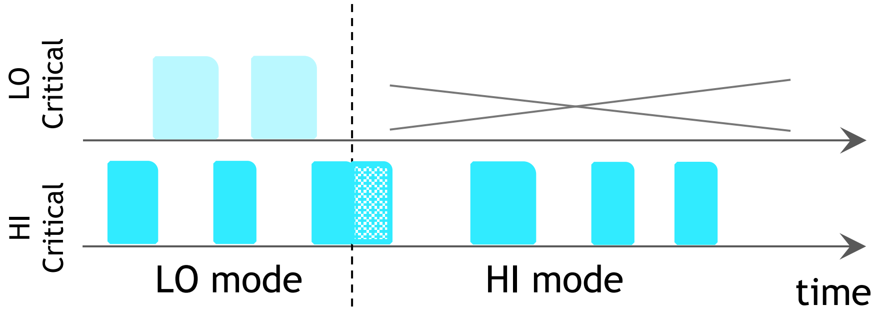
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

apps of varying criticality levels

share a common platform

time overrun is a concern

asymmetric isolation



Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

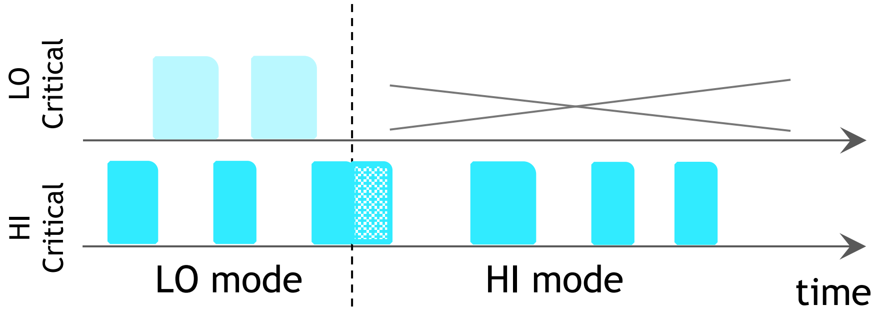
EDF-VD for sporadic tasks (T_i, D_i, $C_i(LO)$, $C_i(HI)$)

Safety preparation

HI tasks' deadlines shortened by a factor x<1 in LO mode

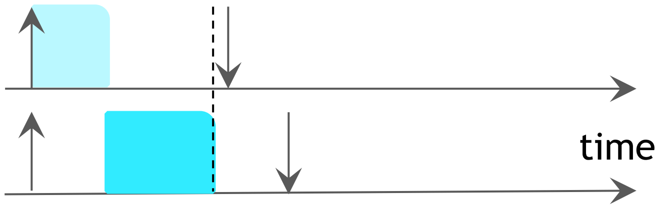
- apps of varying criticality levels
- share a common platform
- time overrun is a concern

- asymmetric isolation



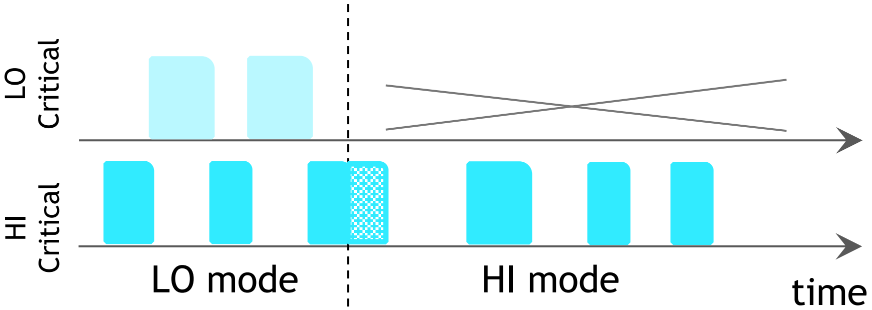
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

- EDF-VD for sporadic tasks (T_i, D_i, $C_i(LO)$, $C_i(HI)$)
- Safety preparation
- HI tasks' deadlines shortened by a factor x<1 in LO mode



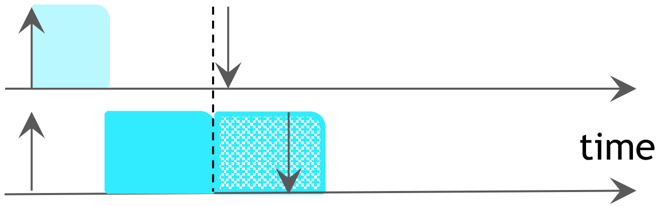
apps of varying criticality levels
share a common platform
time overrun is a concern

asymmetric isolation



Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

EDF-VD for sporadic tasks
$(T_i, D_i, C_i(LO), C_i(HI))$
Safety preparation
HI tasks' deadlines shortened by a factor x<1 in LO mode

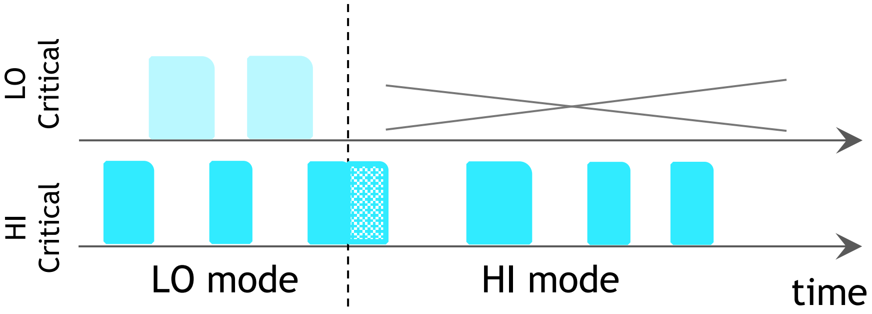


apps of varying criticality levels

share a common platform

time overrun is a concern

asymmetric isolation



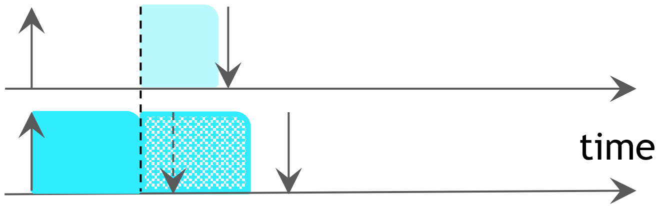
Exploring Energy Saving for Mixed-Criticality Systems on Multi-cores

EDF-VD for sporadic tasks

$(T_i, D_i, C_i(LO), C_i(HI))$

Safety preparation

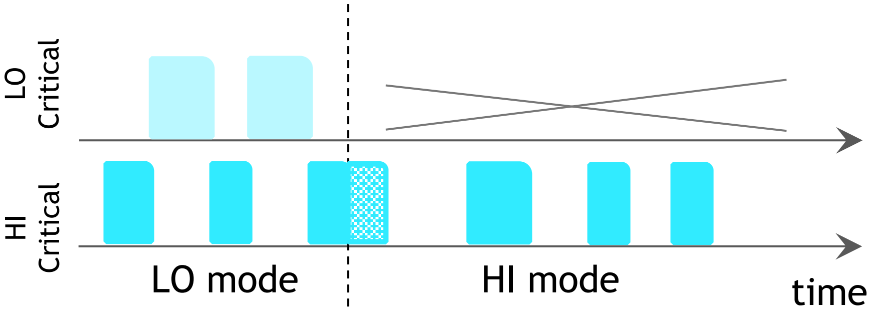
HI tasks' deadlines shortened by a factor x<1 in LO mode



Is it fundamentally different?

Let us focus on unicore first.

Intrinsically Dynamic

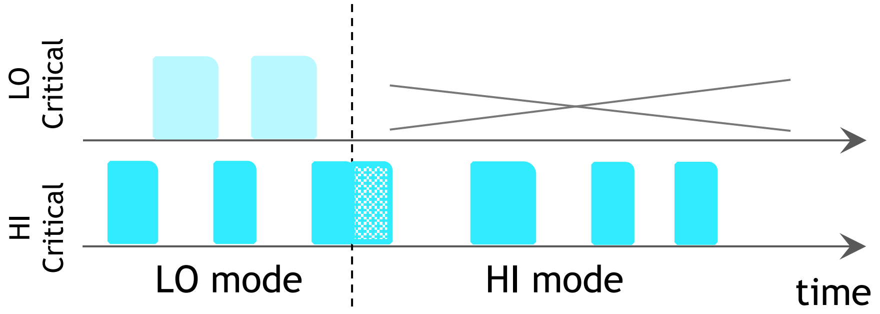


1. For which mode(s) should we minimize energy consumption?

2. Relation between LO and HI mode minimal energy consumptions?

3. Impact of safety preparation?

Intrinsically Dynamic

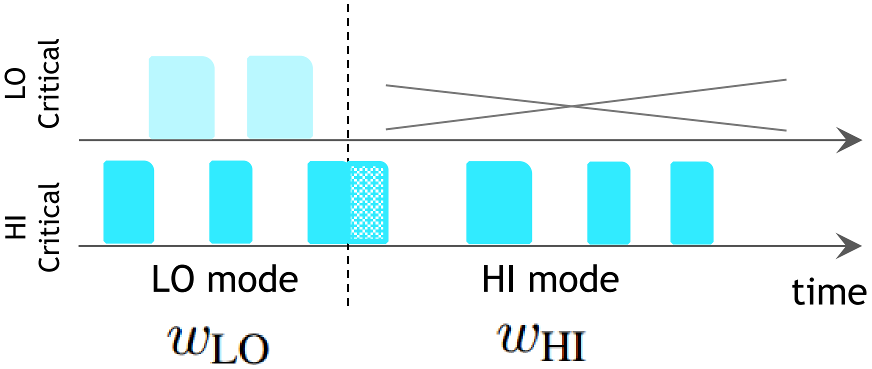


1. For which mode(s) should we minimize energy consumption?

- HI mode is unlikely, hence focus on LO mode[+].
- HI mode probability increases with time, minimize both.

[+] Energy Efficient DVFS Scheduling for Mixed-Criticality Systems, EMSOFT 14

Intrinsically Dynamic



$w_{LO} + w_{HI} = 1$

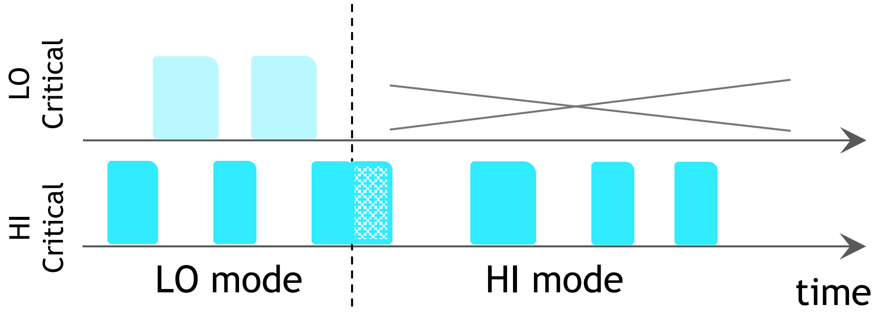
Minimize a weighted sum of energy dissipations in both modes.

$$E_{\mathrm{LO}} = w_{\mathrm{LO}} \sum_{\tau_i \in \tau} \frac{C_i(\mathrm{LO})}{T_i} \frac{f_b}{f_i^{\mathrm{LO}}} (P_s + \beta (f_i^{\mathrm{LO}})^{\alpha})$$

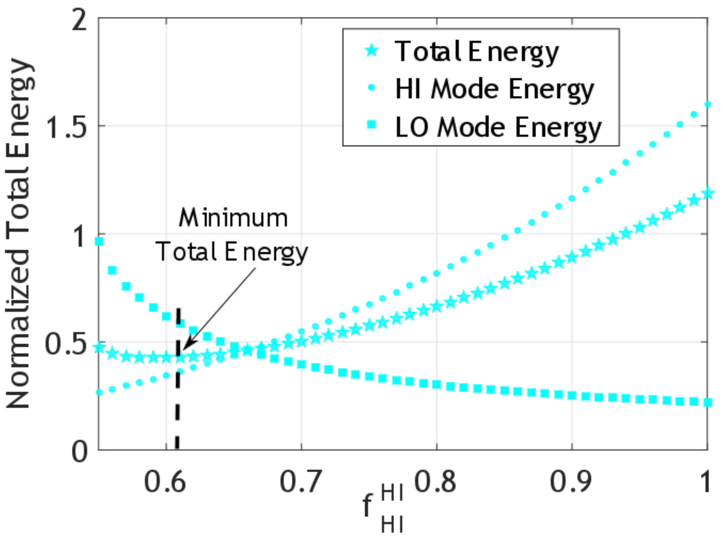
$$E_{\mathrm{HI}} = w_{\mathrm{HI}} \sum_{\tau_i \in \tau_{\mathrm{HI}}} \frac{C_i(\mathrm{HI})}{T_i} \frac{f_b}{f_i^{\mathrm{HI}}} (P_s + \beta (f_i^{\mathrm{HI}})^{\alpha}).$$

$$E = E_{\mathrm{LO}} + E_{\mathrm{HI}}$$

Intrinsically Dynamic



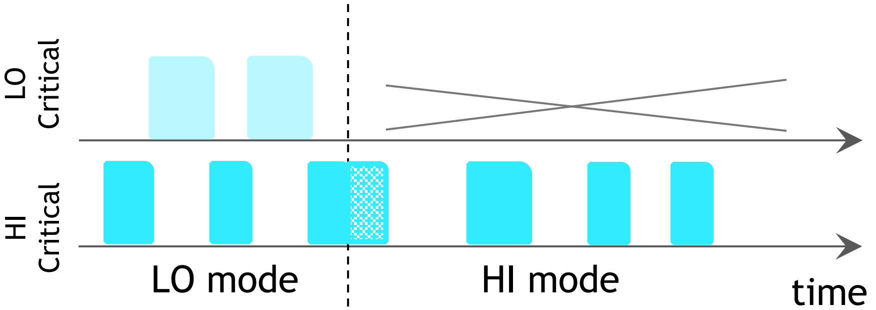
2. Relation between LO and HI mode minimal energy consumptions?



$w_{HI} = 0.65$

Example task set
3 HI, 2 LO

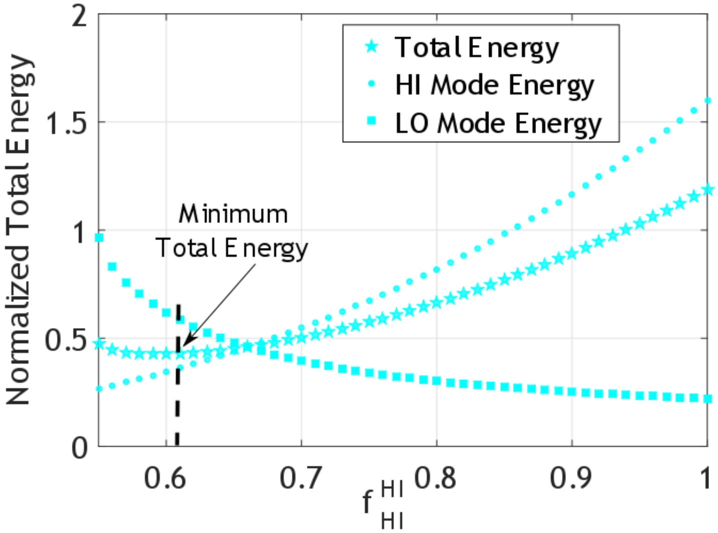
Intrinsically Dynamic



2. Relation between LO and HI mode minimal energy consumptions?

Speedup in HI mode → Less pressure in LO mode

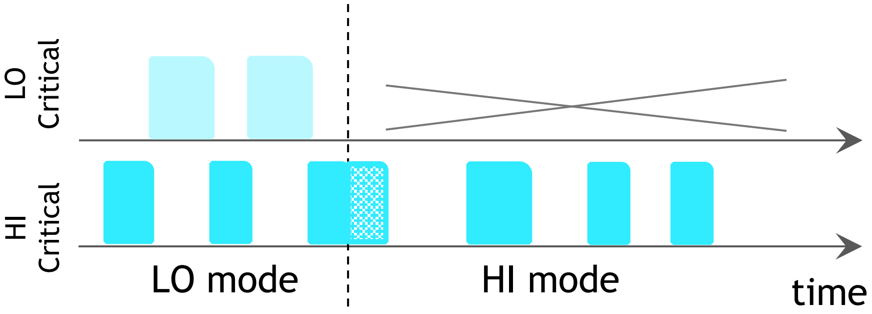
Higher HI energy → Lower LO energy



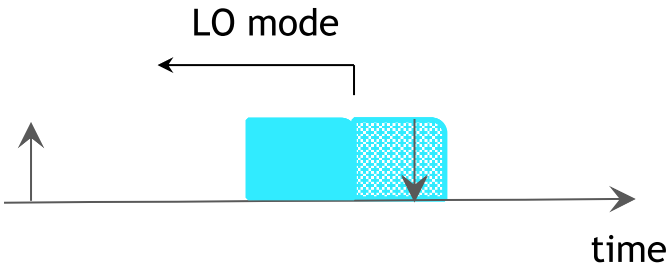
$w_{HI} = 0.65$

Example task set
3 HI, 2 LO

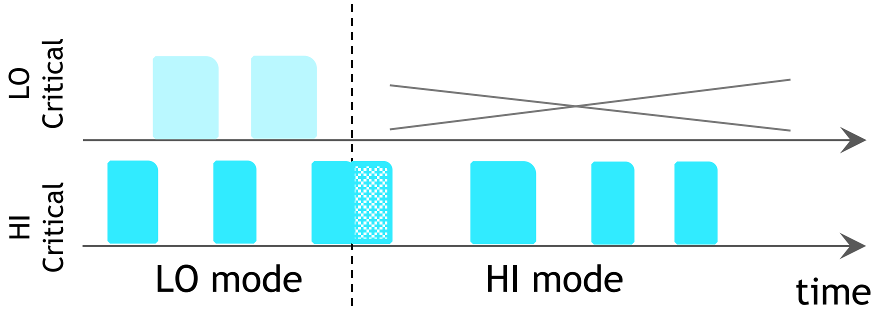
Intrinsically Dynamic



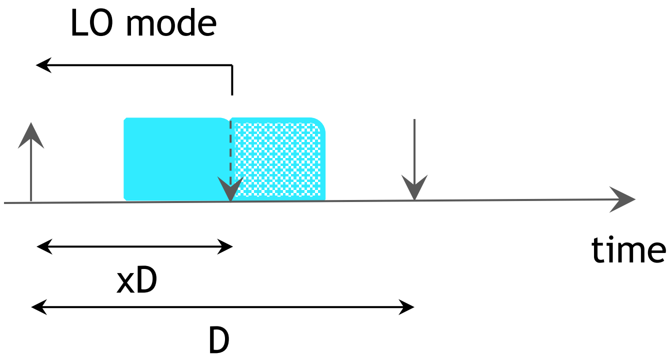
3. Impact of safety preparation?



Intrinsically Dynamic



3. Impact of safety preparation?



More preparation →

Load shifted more to LO mode →

Higher LO mode energy &
Lower HI mode energy

Knowing the characteristics,

how can we solve the unicore problem algorithmically?

The “easy” way

$$
\begin{aligned}
\textbf{minimize} \quad & E = E_{\text{LO}} + E_{\text{HI}} \\
\textbf{s.t.} \quad & \frac{\widetilde{U}_{\text{HI}}^{\text{LO}}}{x} + \widetilde{U}_{\text{LO}}^{\text{LO}} \leq 1 \\
& x\widetilde{U}_{\text{LO}}^{\text{LO}} + \widetilde{U}_{\text{HI}}^{\text{HI}} \leq 1 \\
& x \in [\hat{x}_{\text{LB}}\,,\ \hat{x}_{\text{UB}}] \\
& \forall \tau_i\,,\ \forall \chi\,,\ f_i^{\chi} \in [f_{\text{min}}\,,\ f_{\text{max}}]
\end{aligned}
$$

The "easy" way

$$\begin{aligned}
\textbf{minimize} \quad & E = E_{\text{LO}} + E_{\text{HI}} \\
\textbf{s.t.} \quad & \frac{\widetilde{U}_{\text{HI}}^{\text{LO}}}{x} + \widetilde{U}_{\text{LO}}^{\text{LO}} \leq 1 \\
& x\widetilde{U}_{\text{LO}}^{\text{LO}} + \widetilde{U}_{\text{HI}}^{\text{HI}} \leq 1 \\
& x \in [\hat{x}_{\text{LB}} \, , \, \hat{x}_{\text{UB}}] \\
& \forall \tau_i \, , \forall \chi \, , \, f_i^{\chi} \in [f_{\min} \, , \, f_{\max}]
\end{aligned}$$

LO mode schedulability

HI mode schedulability

feasible preparation

available frequencies

The "easy" way

$$
\begin{aligned}
\textbf{minimize} \quad & E = E_{\text{LO}} + E_{\text{HI}} \\
\textbf{s.t.} \quad & \frac{\widetilde{U}_{\text{HI}}^{\text{LO}}}{x} + \widetilde{U}_{\text{LO}}^{\text{LO}} \leq 1 \\
& x\widetilde{U}_{\text{LO}}^{\text{LO}} + \widetilde{U}_{\text{HI}}^{\text{HI}} \leq 1 \\
& x \in [\hat{x}_{\text{LB}}\,,\ \hat{x}_{\text{UB}}] \\
& \forall \tau_i\,,\ \forall \chi\,,\ f_i^{\chi} \in [f_{\min}\,,\ f_{\max}]
\end{aligned}
$$

LO mode schedulability

HI mode schedulability

feasible preparation

available frequencies

It's a convex programming problem!

The “easy” way

$$\begin{aligned}
\textbf{minimize} \quad & E = E_{\text{LO}} + E_{\text{HI}} \\
\textbf{s.t.} \quad & \frac{\widetilde{U}_{\text{HI}}^{\text{LO}}}{x} + \widetilde{U}_{\text{LO}}^{\text{LO}} \le 1 \\
& x\widetilde{U}_{\text{LO}}^{\text{LO}} + \widetilde{U}_{\text{HI}}^{\text{HI}} \le 1 \\
& x \in [\hat{x}_{\text{LB}} \, , \, \hat{x}_{\text{UB}}] \\
& \forall \tau_i \, , \forall \chi \, , \, f_i^{\chi} \in [f_{\min} \, , \, f_{\max}]
\end{aligned}$$

LO mode schedulability

HI mode schedulability

feasible preparation

available frequencies

It’s a convex programming problem!

Even “better”, one can apply KKT conditions!

The "easy" way

$$
\begin{aligned}
\textbf{minimize} \quad & E = E_{\mathrm{LO}} + E_{\mathrm{HI}} \\
\textbf{s.t.} \quad & \frac{\widetilde{U}_{\mathrm{HI}}^{\mathrm{LO}}}{x} + \widetilde{U}_{\mathrm{LO}}^{\mathrm{LO}} \leq 1 \\
& x\widetilde{U}_{\mathrm{LO}}^{\mathrm{LO}} + \widetilde{U}_{\mathrm{HI}}^{\mathrm{HI}} \leq 1 \\
& x \in [\hat{x}_{\mathrm{LB}}\,,\, \hat{x}_{\mathrm{UB}}] \\
& \forall \tau_i\,,\, \forall \chi\,,\, f_i^{\chi} \in [f_{\mathrm{min}}\,,\, f_{\mathrm{max}}]
\end{aligned}
$$

LO mode schedulability

HI mode schedulability

feasible preparation

available frequencies

It's a convex programming problem!

Even "better", one can apply KKT conditions!

Lagrange multiplier method extended for inequality

Is that all for unicore?

What are good insights leading to a faster algorithmic solution?

Reduce frequency search space

$$\forall \tau_i \in \tau_{\mathrm{LO}}, f_i^{\mathrm{LO}} = f_{\mathrm{LO}}^{\mathrm{LO}}; \quad \forall \tau_i \in \tau_{\mathrm{HI}}, f_i^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} \wedge f_i^{\mathrm{HI}} = f_{\mathrm{HI}}^{\mathrm{HI}}$$

Reduce frequency search space

$$\forall \tau_i \in \tau_{\mathrm{LO}}, f_i^{\mathrm{LO}} = f_{\mathrm{LO}}^{\mathrm{LO}}; \quad \forall \tau_i \in \tau_{\mathrm{HI}}, f_i^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} \wedge f_i^{\mathrm{HI}} = f_{\mathrm{HI}}^{\mathrm{HI}}$$

consider τ_i and τ_j

minimize

$$E_{\mathrm{HI}} = w_{\mathrm{HI}} \left(u_i \frac{f_b}{f_i^{\mathrm{HI}}} \left(P_s + \beta (f_i^{\mathrm{HI}})^{\alpha}\right) \right) + w_{\mathrm{HI}} \left(u_j \frac{f_b}{f_j^{\mathrm{HI}}} \left(P_s + \beta (f_j^{\mathrm{HI}})^{\alpha}\right) \right)$$

s.t. $u_i \frac{f_b}{f_i^{\mathrm{HI}}} + u_j \frac{f_b}{f_j^{\mathrm{HI}}} \leq u_{i+j}$

Reduce frequency search space

$$\forall \tau_i \in \tau_{\mathrm{LO}}, f_i^{\mathrm{LO}} = f_{\mathrm{LO}}^{\mathrm{LO}}; \quad \forall \tau_i \in \tau_{\mathrm{HI}}, f_i^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} \wedge f_i^{\mathrm{HI}} = f_{\mathrm{HI}}^{\mathrm{HI}}$$

consider τ_i and τ_j

minimize

$$E_{\mathrm{HI}} = w_{\mathrm{HI}} \left(u_i \frac{f_b}{f_i^{\mathrm{HI}}} \left(P_s + \beta (f_i^{\mathrm{HI}})^{\alpha}\right) \right) + w_{\mathrm{HI}} \left(u_j \frac{f_b}{f_j^{\mathrm{HI}}} \left(P_s + \beta (f_j^{\mathrm{HI}})^{\alpha}\right) \right)$$

s.t. $u_i \frac{f_b}{f_i^{\mathrm{HI}}} + u_j \frac{f_b}{f_j^{\mathrm{HI}}} \leq u_{i+j}$

Reduce frequency search space

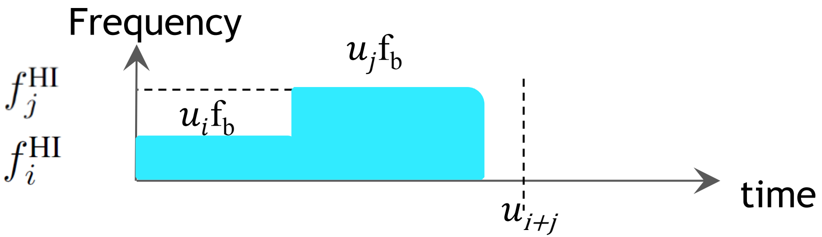
$$\forall \tau_i \in \tau_{\text{LO}}, f_i^{\text{LO}} = f_{\text{LO}}^{\text{LO}}; \quad \forall \tau_i \in \tau_{\text{HI}}, f_i^{\text{LO}} = f_{\text{HI}}^{\text{LO}} \wedge f_i^{\text{HI}} = f_{\text{HI}}^{\text{HI}}$$

consider τ_i and τ_j

minimize

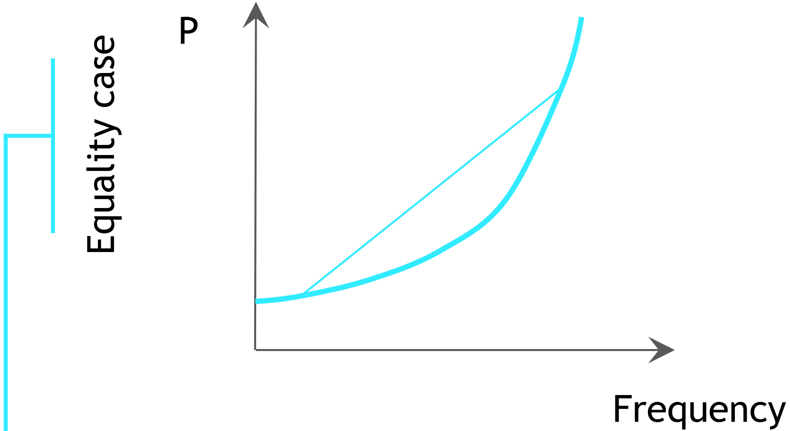
$$E_{\text{HI}} = w_{\text{HI}}\left(u_i \frac{f_b}{f_i^{\text{HI}}}\left(P_s + \beta (f_i^{\text{HI}})^{\alpha}\right)\right) + w_{\text{HI}}\left(u_j \frac{f_b}{f_j^{\text{HI}}}\left(P_s + \beta (f_j^{\text{HI}})^{\alpha}\right)\right)$$

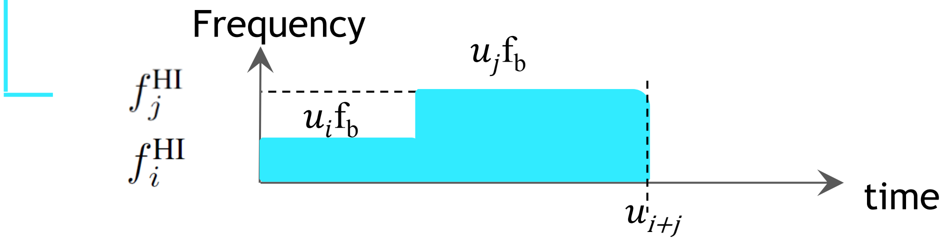
s.t. $u_i \frac{f_b}{f_i^{\text{HI}}} + u_j \frac{f_b}{f_j^{\text{HI}}} \leq u_{i+j}$



Reduce frequency search space

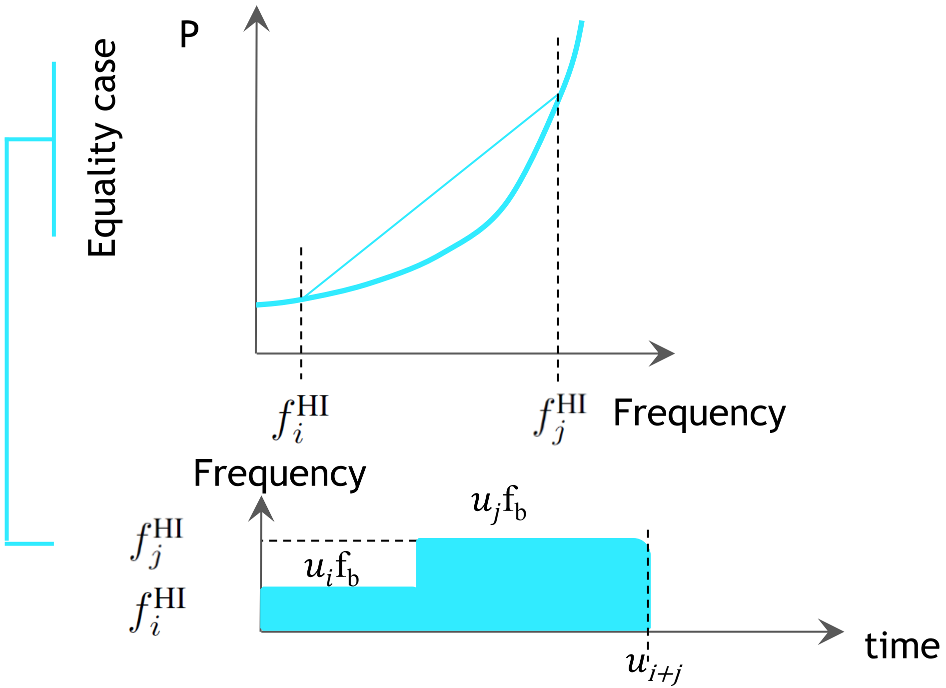
$$\forall \tau_i \in \tau_{\text{LO}}, f_i^{\text{LO}} = f_{\text{LO}}^{\text{LO}}; \quad \forall \tau_i \in \tau_{\text{HI}}, f_i^{\text{LO}} = f_{\text{HI}}^{\text{LO}} \wedge f_i^{\text{HI}} = f_{\text{HI}}^{\text{HI}}$$





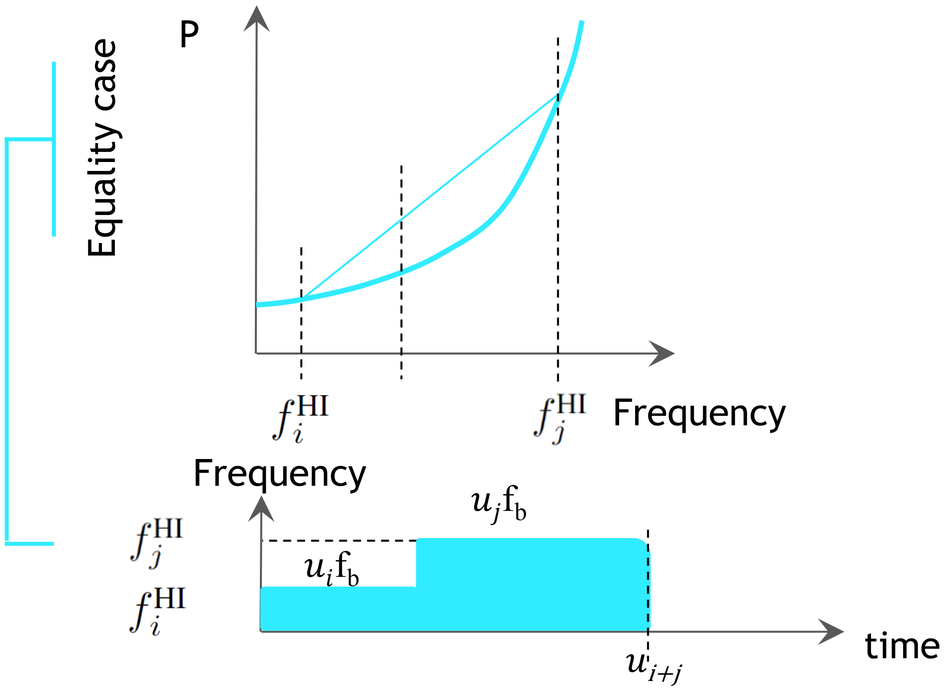
Reduce frequency search space

$$\forall \tau_i \in \tau_{\text{LO}}, f_i^{\text{LO}} = f_{\text{LO}}^{\text{LO}}; \quad \forall \tau_i \in \tau_{\text{HI}}, f_i^{\text{LO}} = f_{\text{HI}}^{\text{LO}} \wedge f_i^{\text{HI}} = f_{\text{HI}}^{\text{HI}}$$



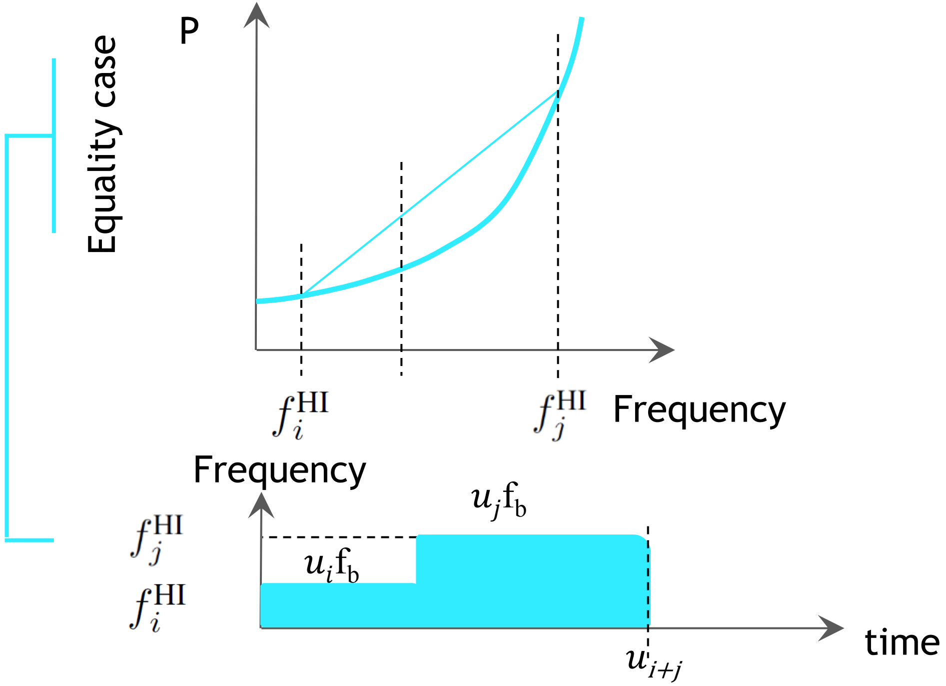
Reduce frequency search space

$$\forall \tau_i \in \tau_{LO}, f_i^{LO} = f_{LO}^{LO}; \quad \forall \tau_i \in \tau_{HI}, f_i^{LO} = f_{HI}^{LO} \wedge f_i^{HI} = f_{HI}^{HI}$$



Reduce frequency search space

$$\forall \tau_i \in \tau_{\mathrm{LO}}, f_i^{\mathrm{LO}} = f_{\mathrm{LO}}^{\mathrm{LO}}; \quad \forall \tau_i \in \tau_{\mathrm{HI}}, f_i^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} \wedge f_i^{\mathrm{HI}} = f_{\mathrm{HI}}^{\mathrm{HI}}$$



General case proved by KKT

“Optimal” safety preparation

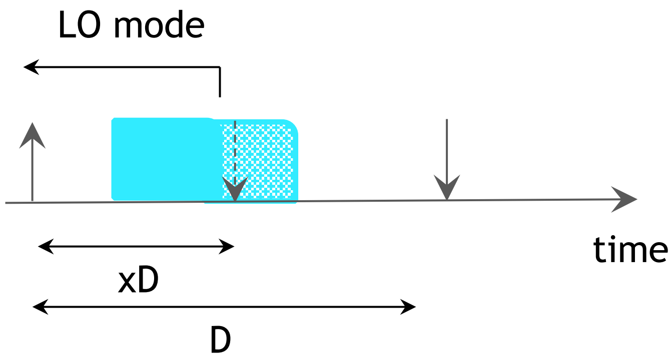
1 Extreme case, i.e. $f_{\mathrm{LO}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{HI}} = f_{\min}$.

2 Equilibrium case, i.e. $x_{\mathrm{LBopt}} = x_{\mathrm{opt}} = x_{\mathrm{UBopt}}$.

“Optimal” safety preparation

1 Extreme case, i.e. $f_{\text{LO}}^{\text{LO}} = f_{\text{HI}}^{\text{LO}} = f_{\text{HI}}^{\text{HI}} = f_{\text{min}}$.

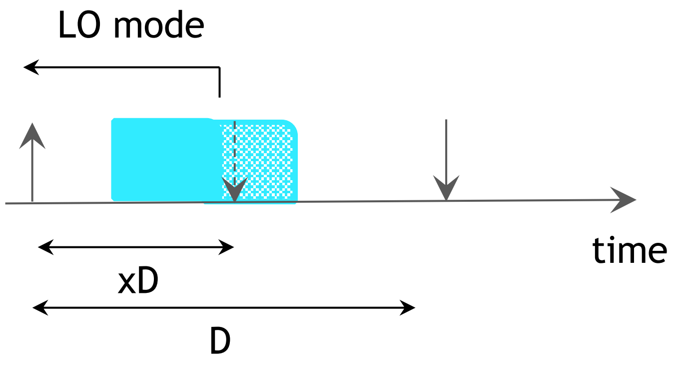
2 Equilibrium case, i.e. $x_{\text{LBopt}} = x_{\text{opt}} = x_{\text{UBopt}}$.

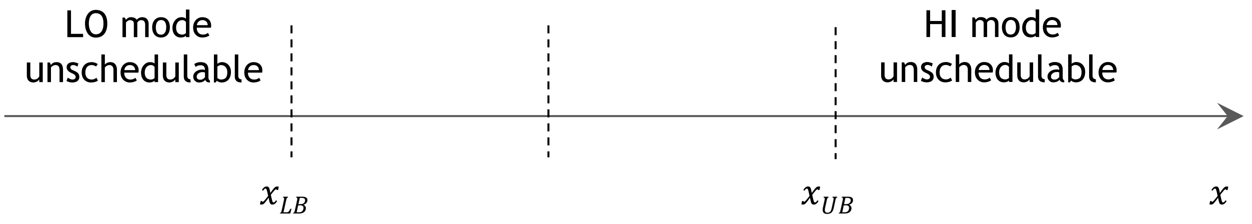


"Optimal" safety preparation

1 Extreme case, i.e. $f_{\mathrm{LO}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{HI}} = f_{\min}$.

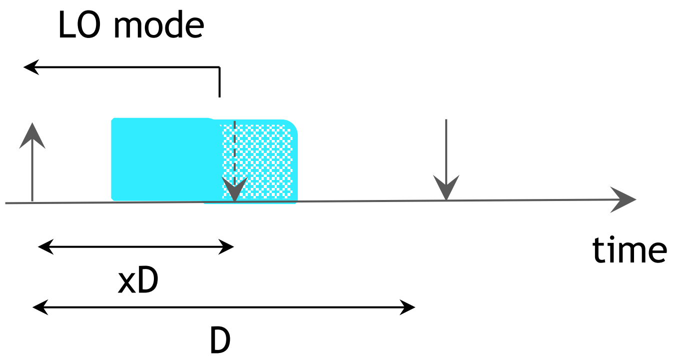
2 Equilibrium case, i.e. $x_{\mathrm{LB\,opt}} = x_{\mathrm{opt}} = x_{\mathrm{UB\,opt}}$.

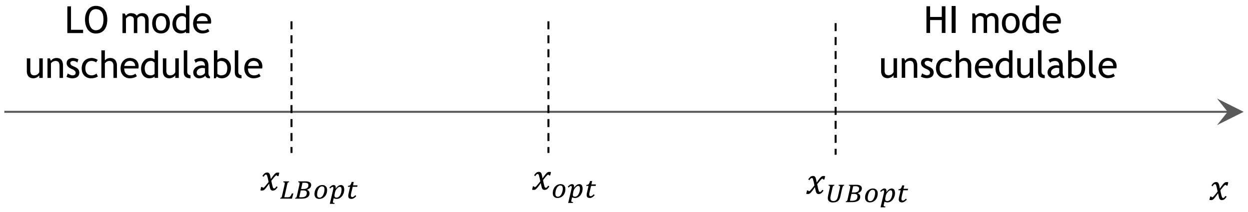




“Optimal” safety preparation

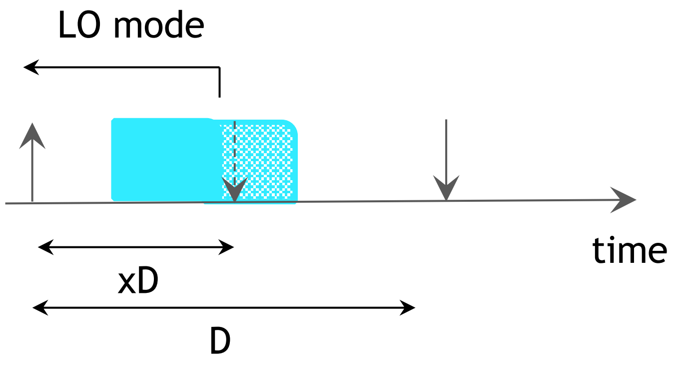
1 Extreme case, i.e. $f_{\mathrm{LO}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{HI}} = f_{\min}$.
2 Equilibrium case, i.e. $x_{\mathrm{LBopt}} = x_{\mathrm{opt}} = x_{\mathrm{UBopt}}$.

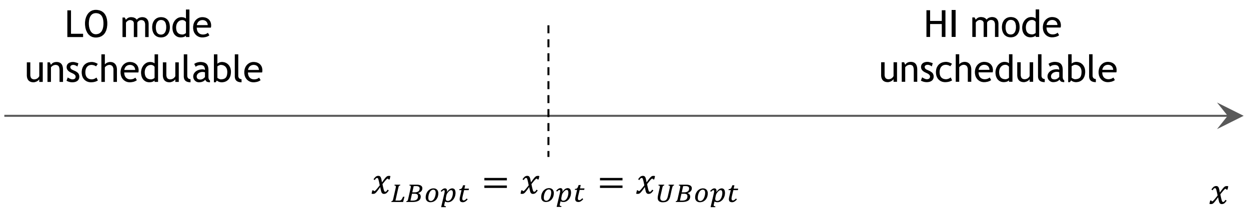




“Optimal” safety preparation

1. Extreme case, i.e. $f_{\mathrm{LO}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{LO}} = f_{\mathrm{HI}}^{\mathrm{HI}} = f_{\min}$.
2. Equilibrium case, i.e. $x_{\mathrm{LBopt}} = x_{\mathrm{opt}} = x_{\mathrm{UBopt}}$.





Reduce frequency search space

“Optimal” safety preparation

Numerical solution to the optimization problem with logarithmic complexity

Multicore

Let us focus on partitioned scheduling first.

Multicore

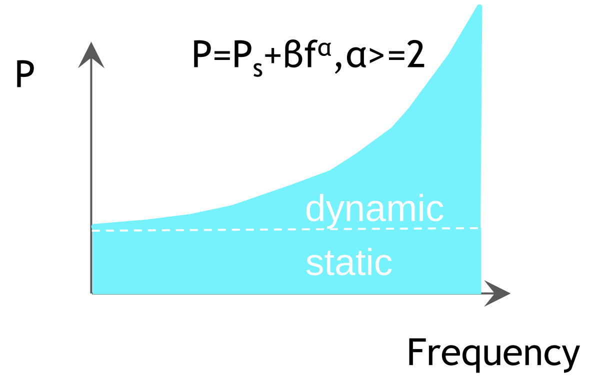
Let us focus on partitioned scheduling first.

We only need to solve the partitioning.

A general intuition

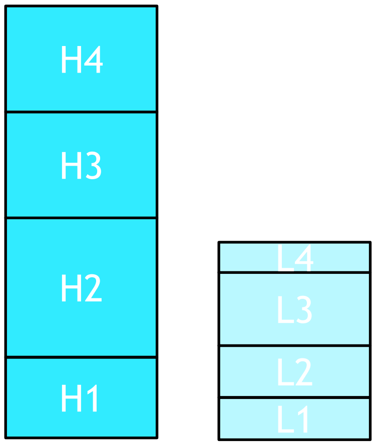
Cores should have equal chances to “relax”.

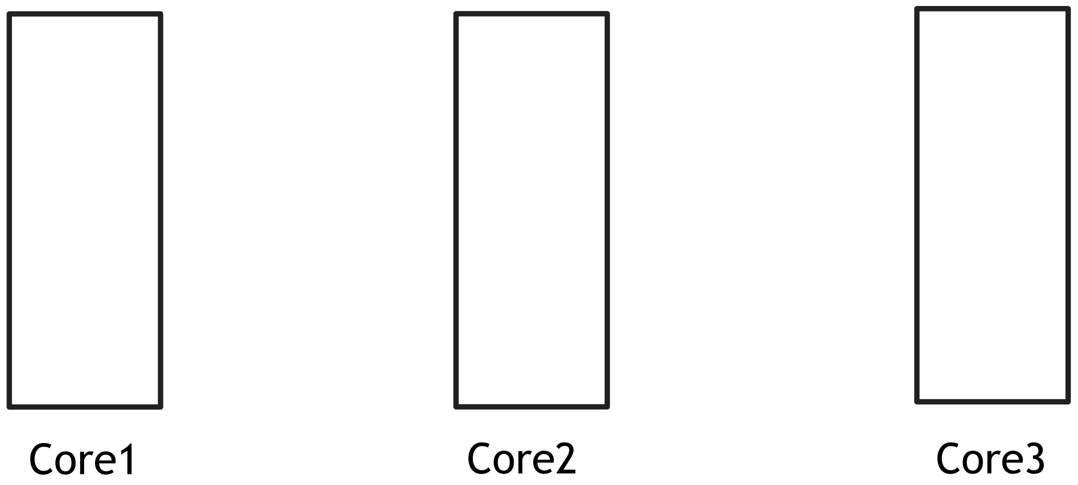
Energy is saved mostly through the “initial” phase.



How good are existing methods?

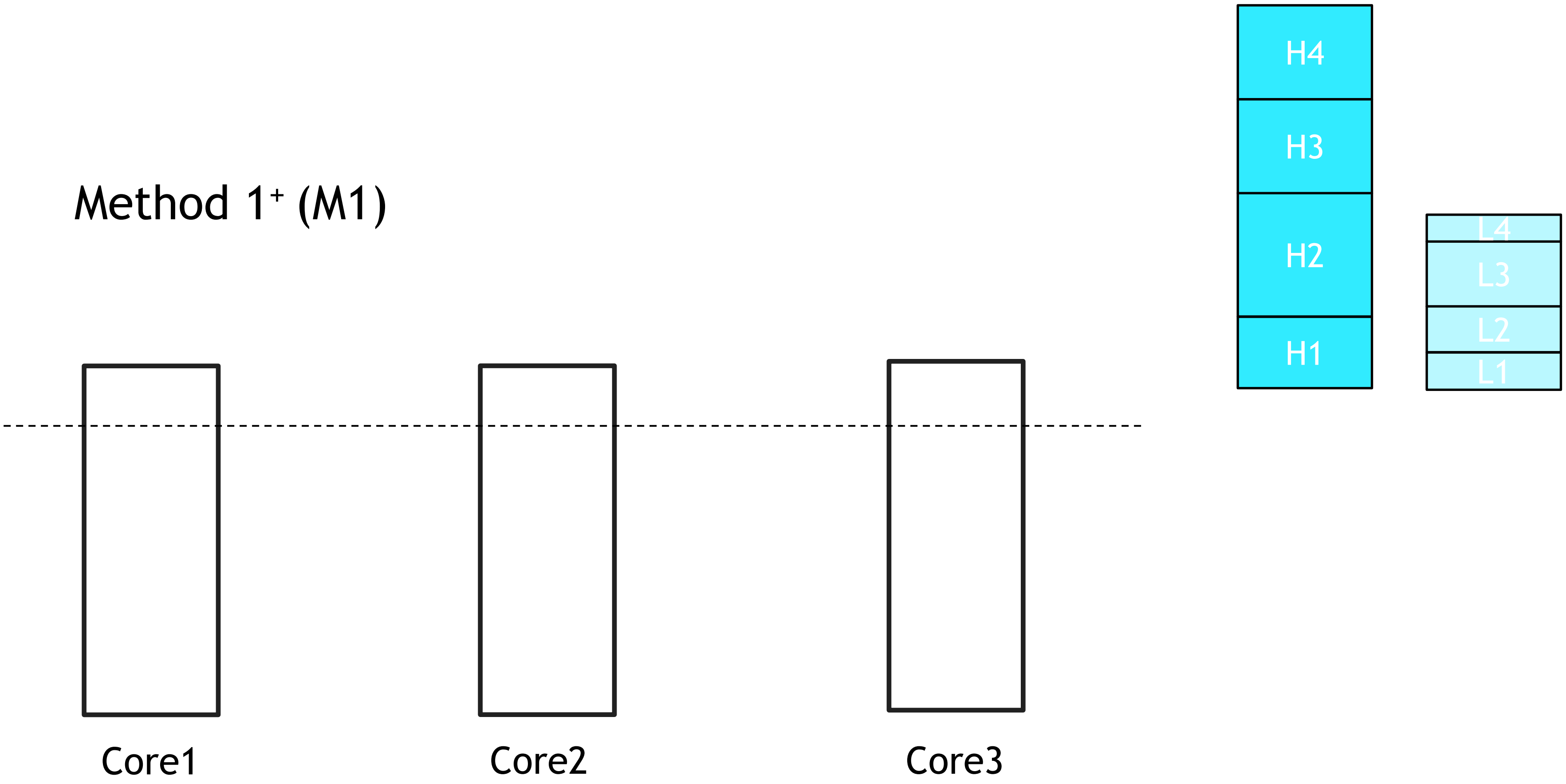
Method 1[+] (M1)





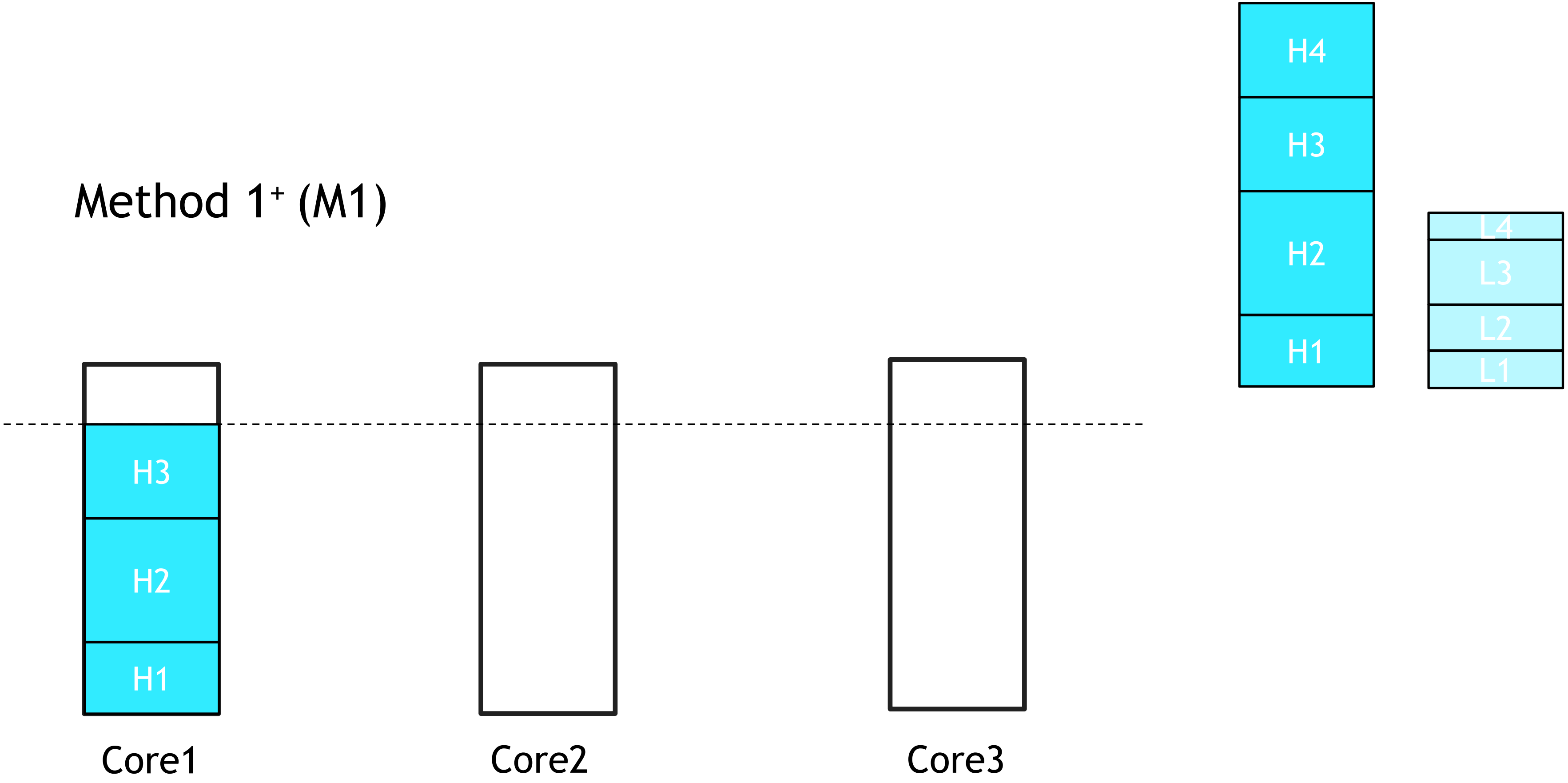
[+]Baruah et al. Mixed-Criticality Scheduling on Multiprocessors, RTS 14

Method 1[+] (M1)



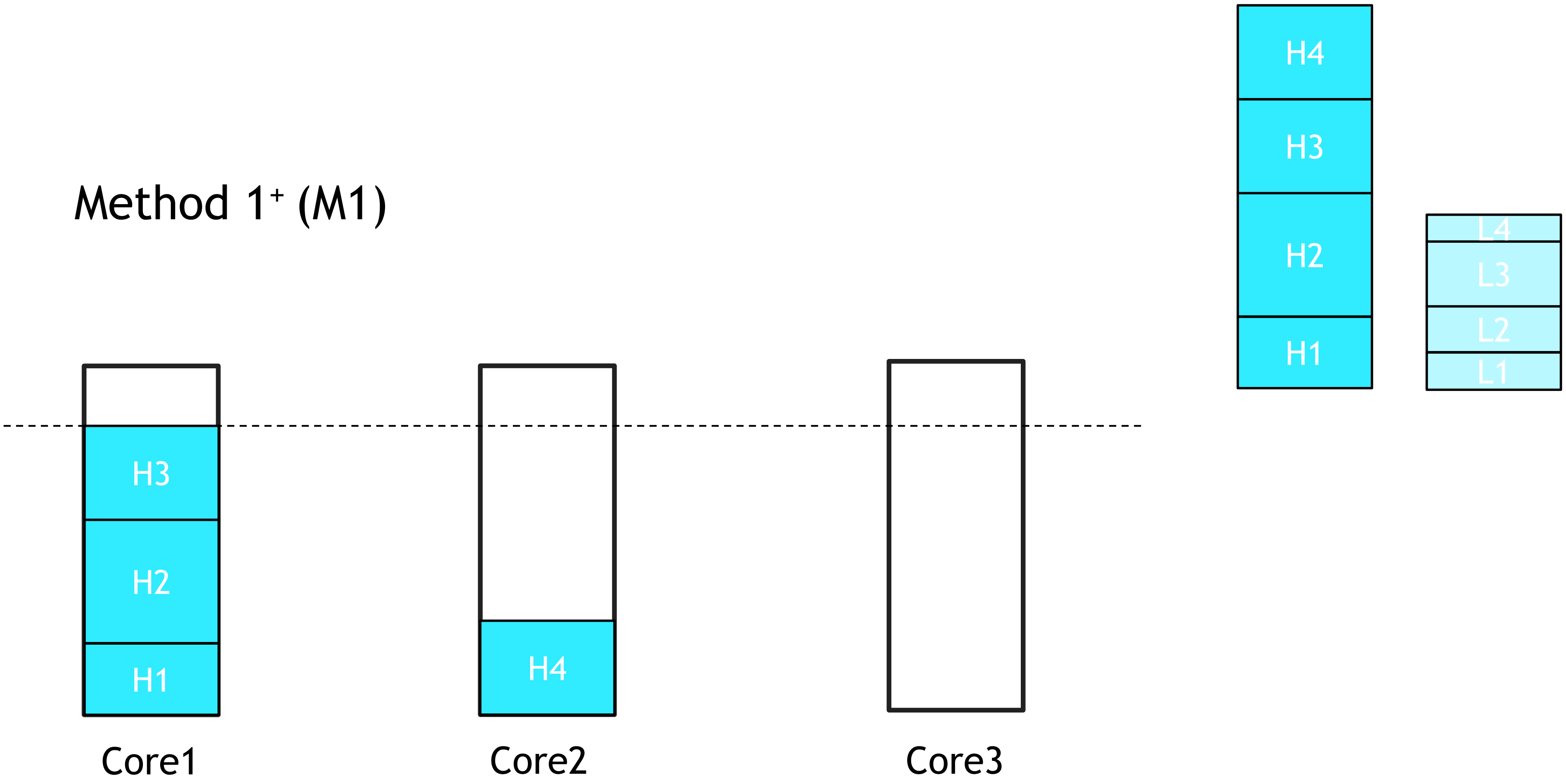
[+]Baruah et al. Mixed-Criticality Scheduling on Multiprocessors, RTS 14

Method 1[+] (M1)



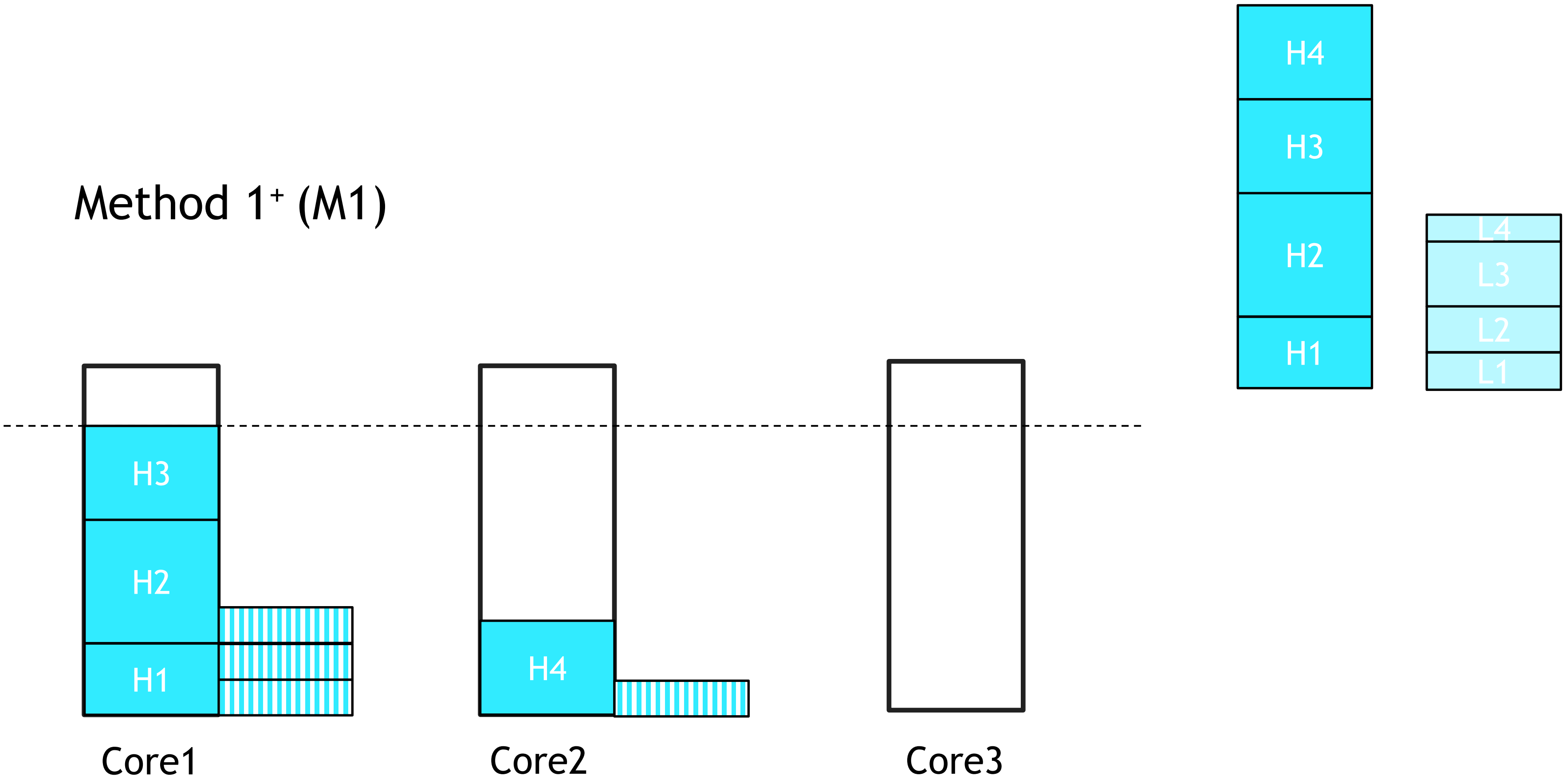
[+]Baruah et al. Mixed-Criticality Scheduling on Multiprocessors, RTS 14

Method 1⁺ (M1)



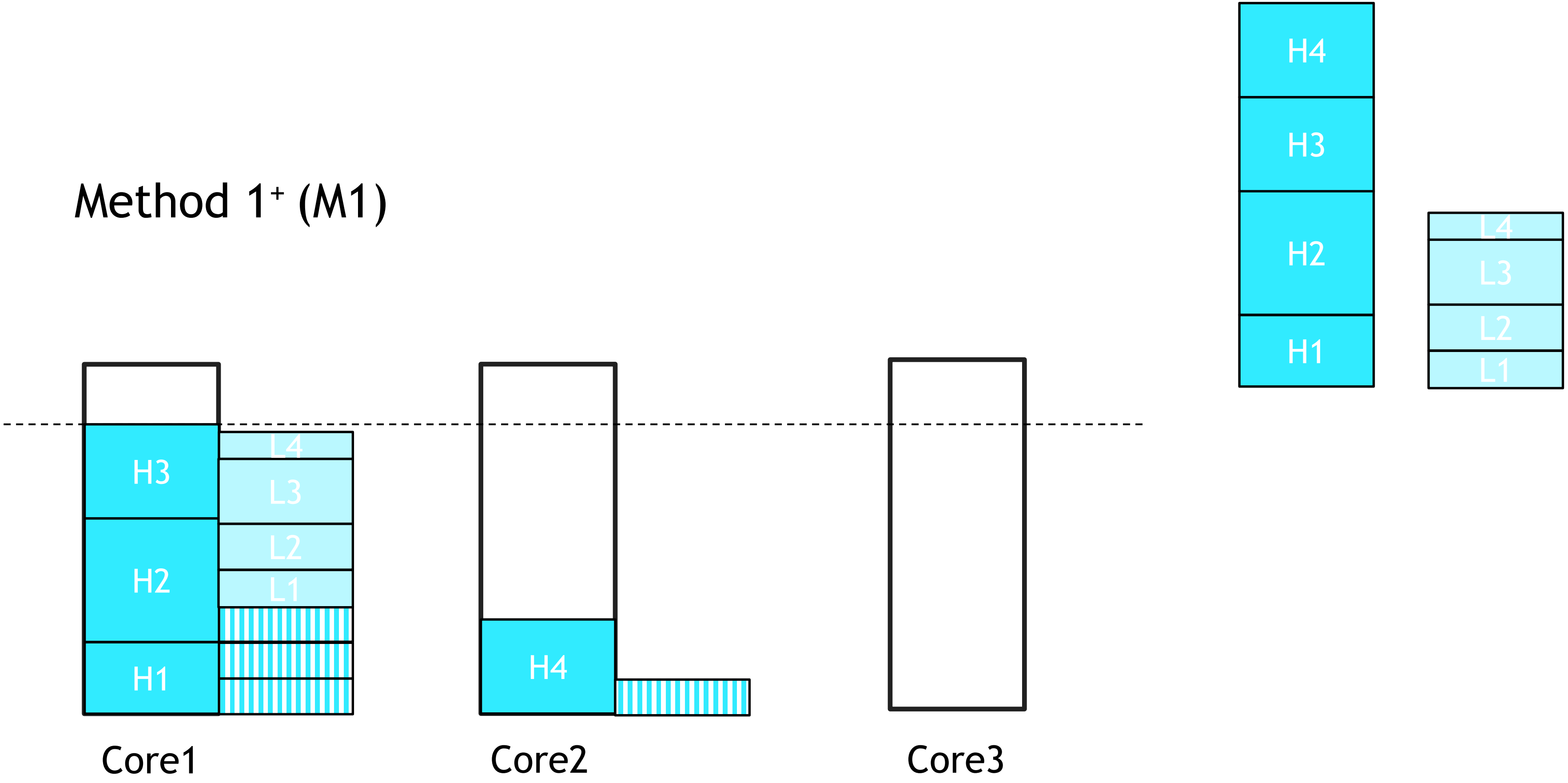
⁺Baruah et al. Mixed-Criticality Scheduling on Multiprocessors, RTS 14

Method 1+ (M1)



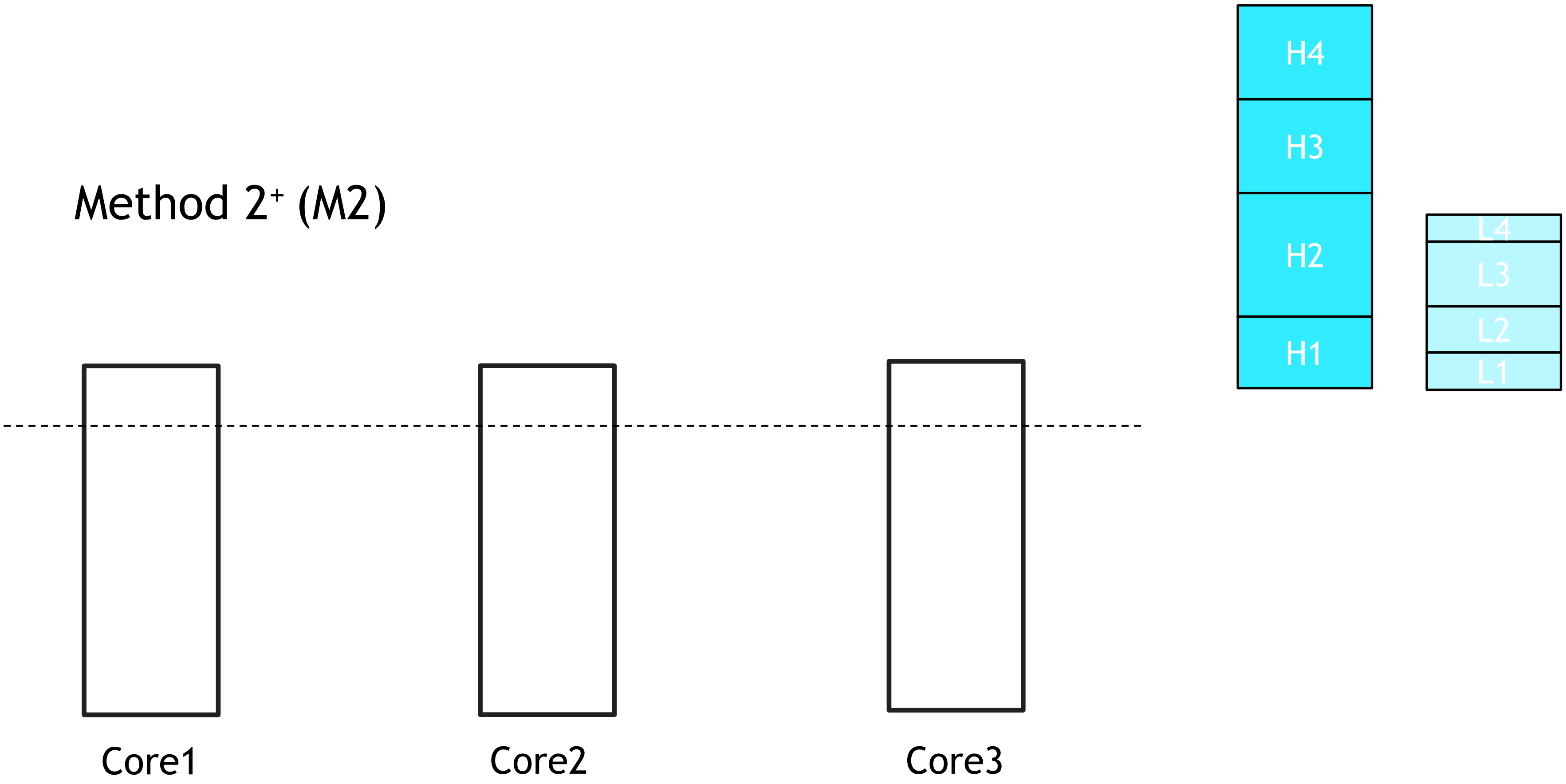
+Baruah et al. Mixed-Criticality Scheduling on Multiprocessors, RTS 14

Method 1⁺ (M1)



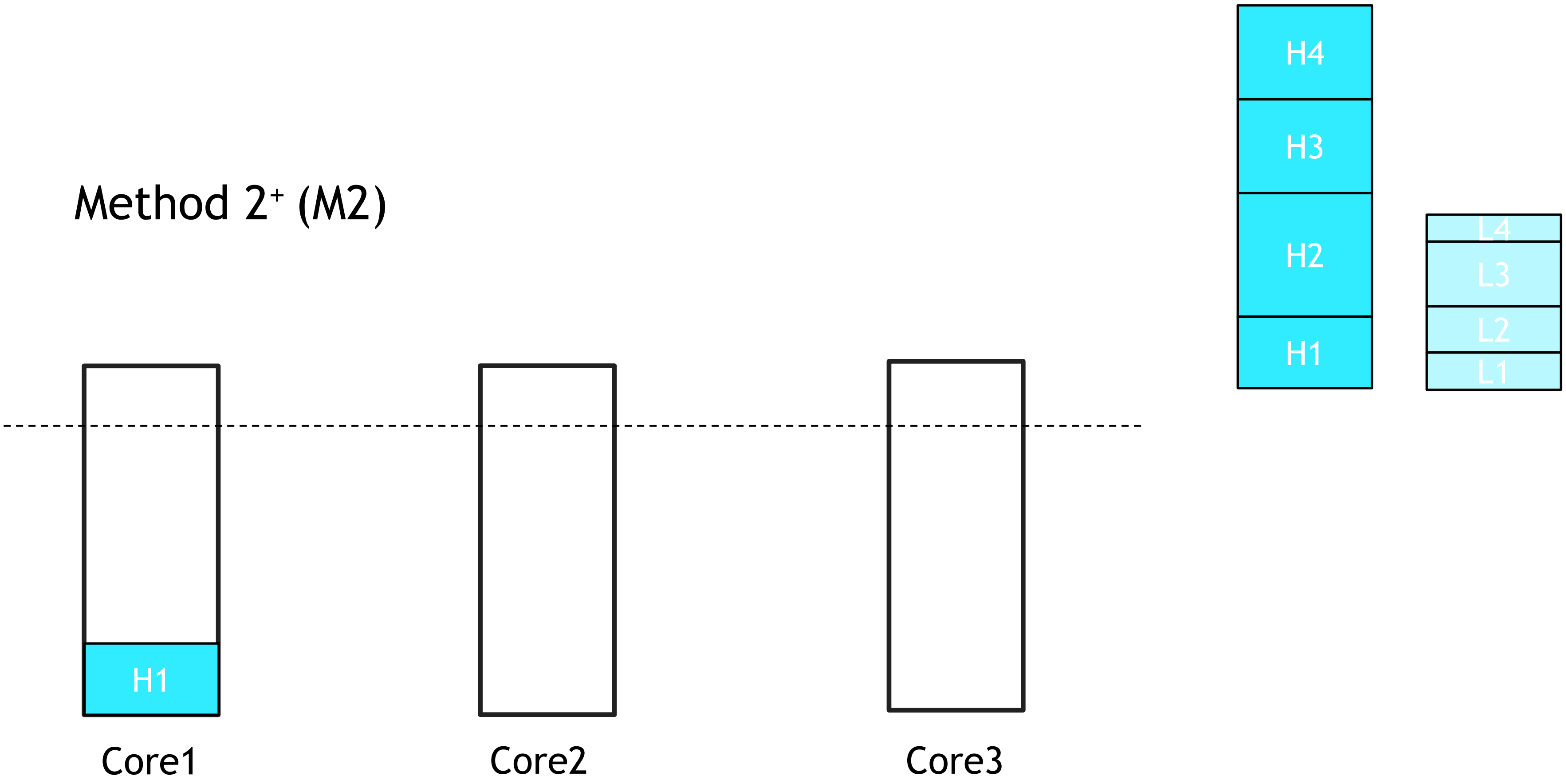
⁺Baruah et al. Mixed-Criticality Scheduling on Multiprocessors, RTS 14

Method 2+ (M2)



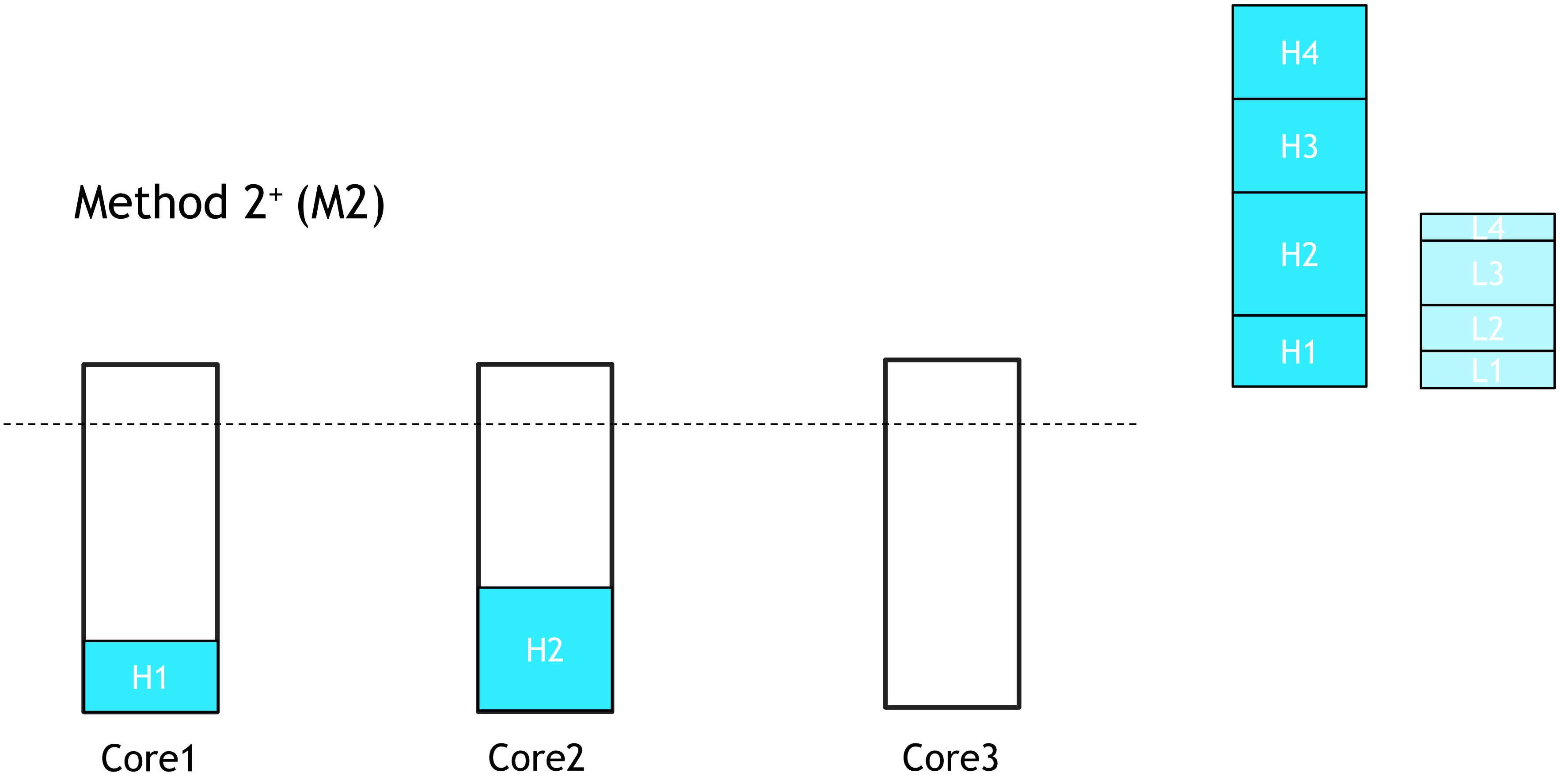
+Gu et al. Partitioned Scheduling of Sporadic Mixed-Criticality Systems on Multiprocessor Platforms, DATE 14

Method 2+ (M2)



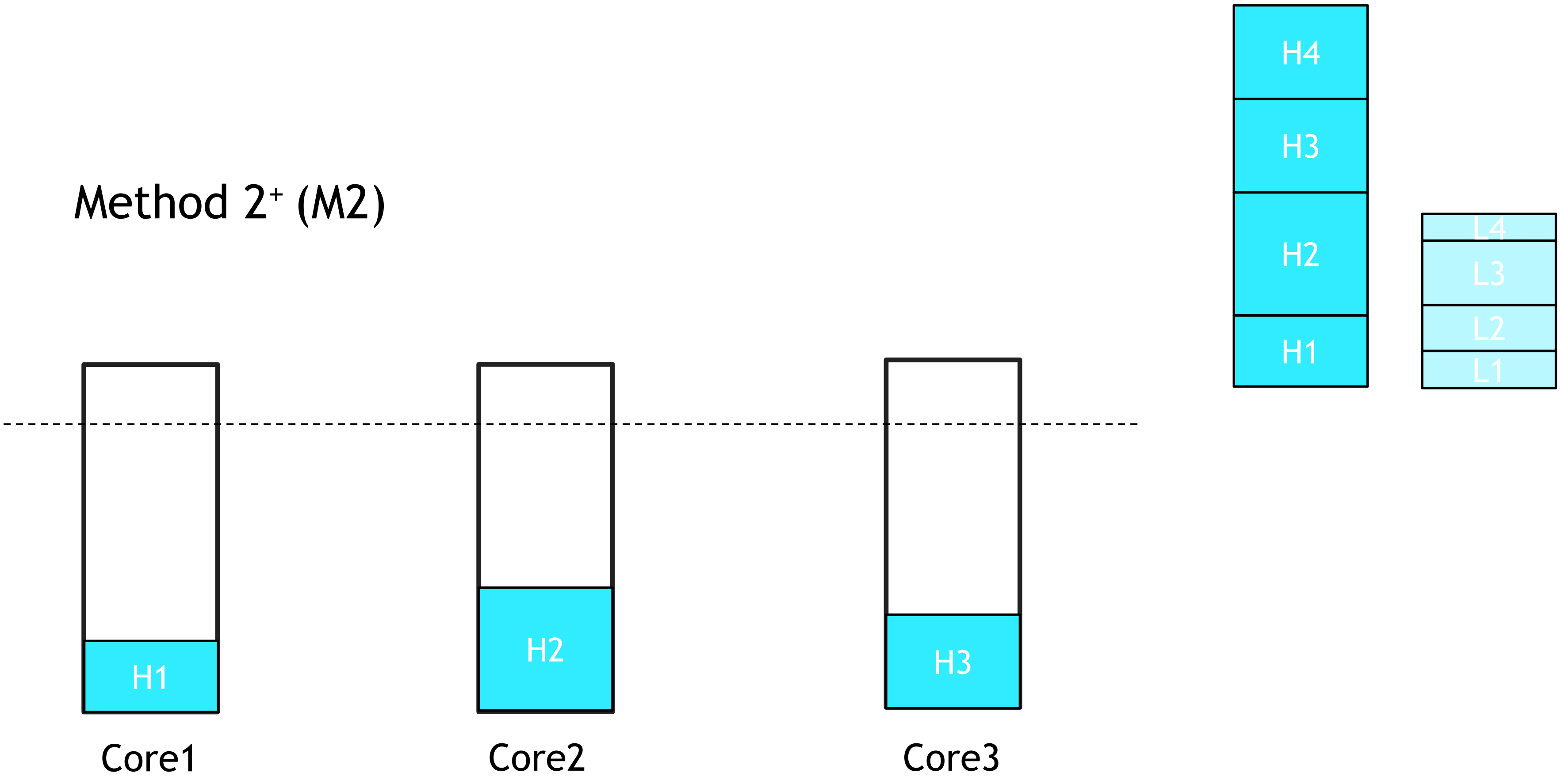
+Gu et al. Partitioned Scheduling of Sporadic Mixed-Criticality Systems on Multiprocessor Platforms, DATE 14

Method 2⁺ (M2)



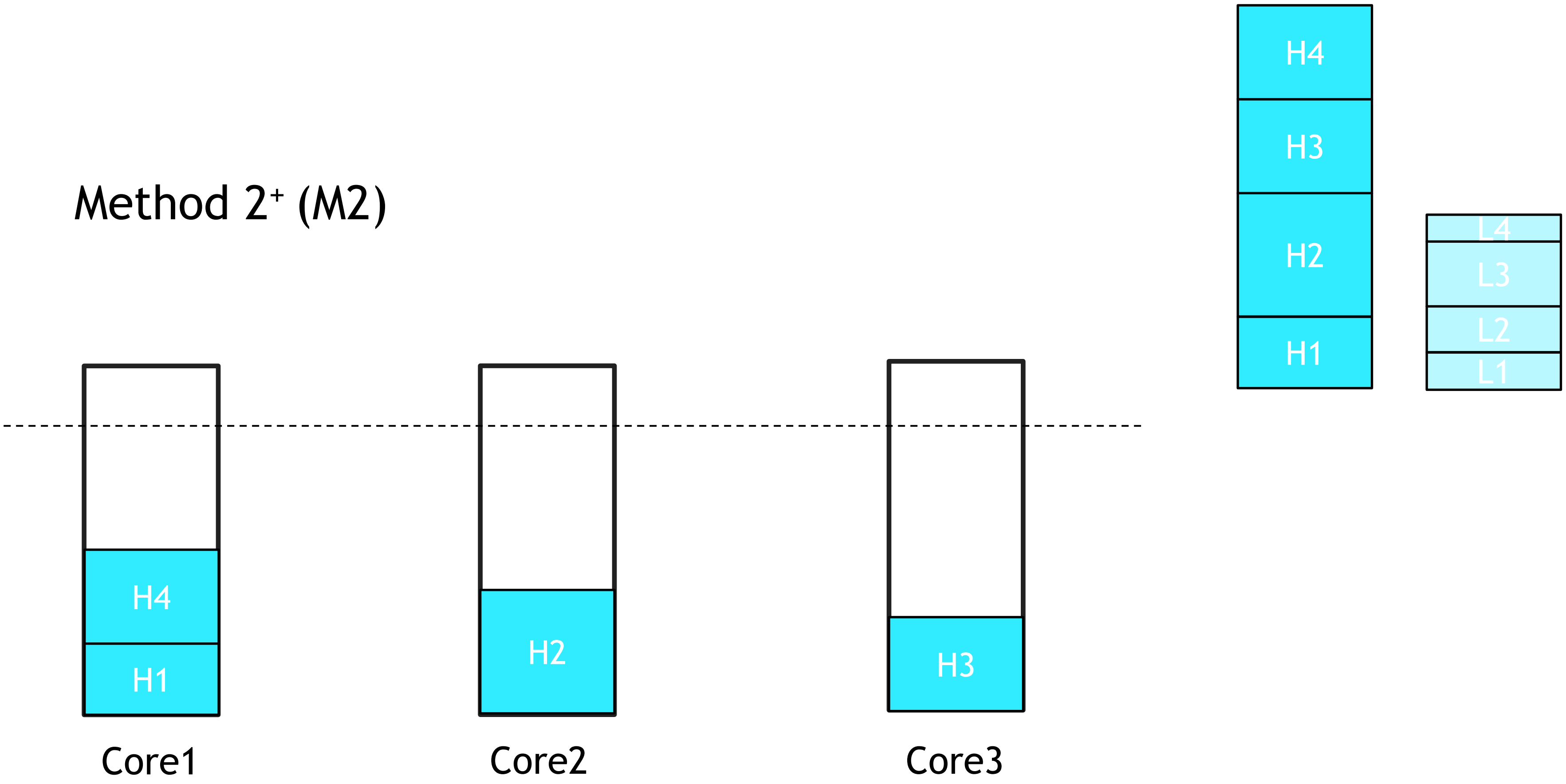
⁺Gu et al. Partitioned Scheduling of Sporadic Mixed-Criticality Systems on Multiprocessor Platforms, DATE 14

Method 2+ (M2)



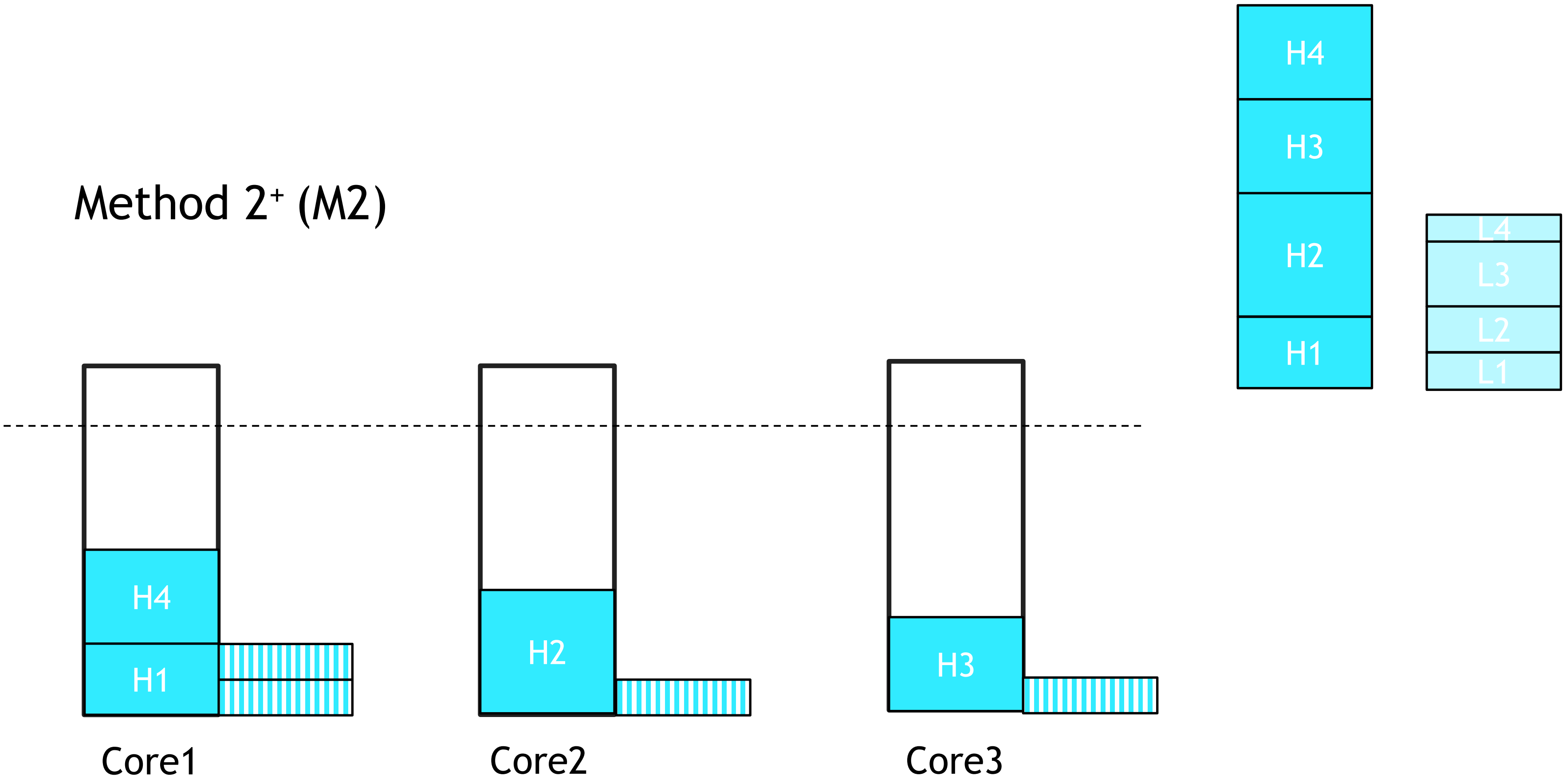
+Gu et al. Partitioned Scheduling of Sporadic Mixed-Criticality Systems on Multiprocessor Platforms, DATE 14

Method 2+ (M2)



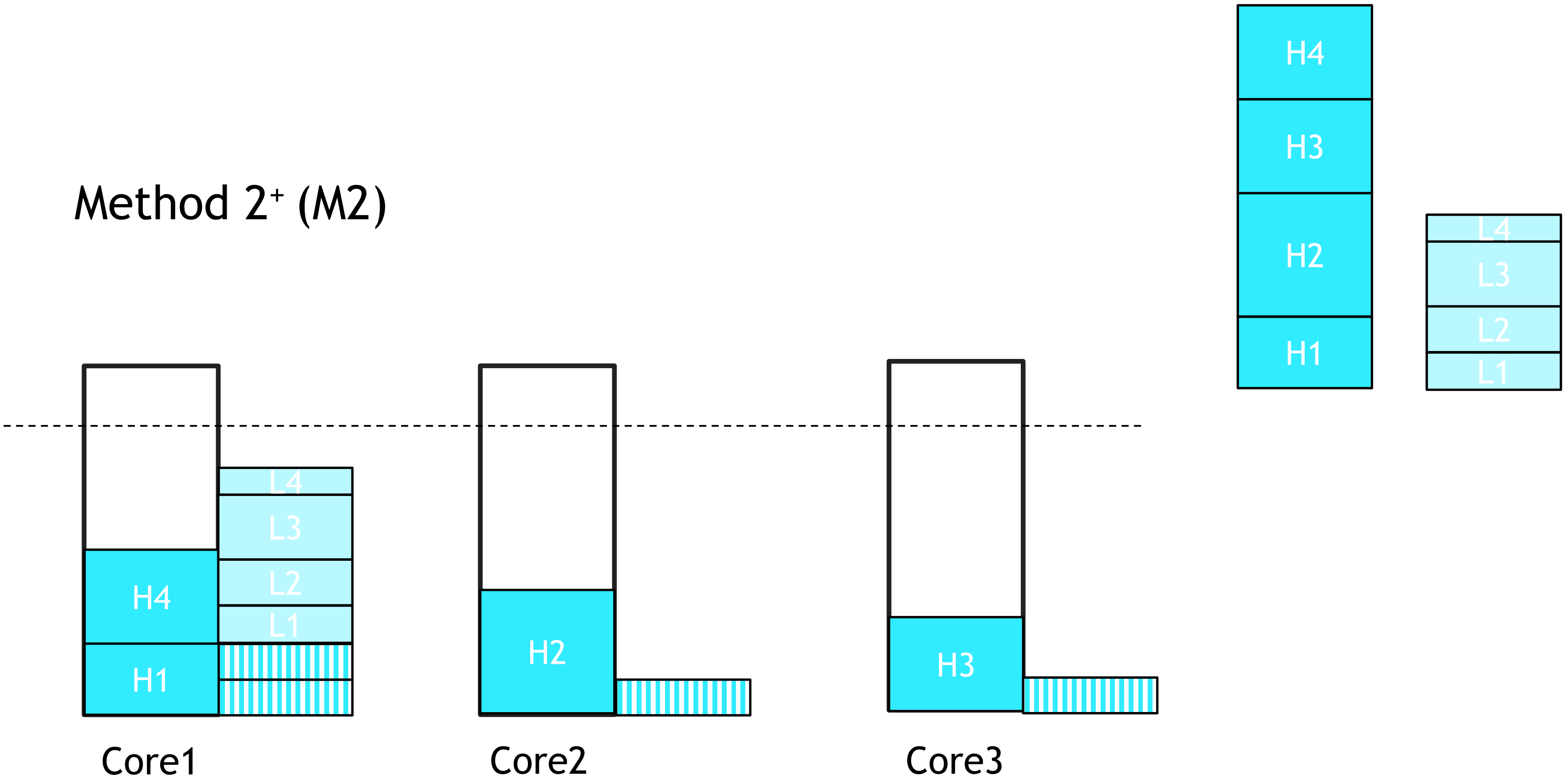
+Gu et al. Partitioned Scheduling of Sporadic Mixed-Criticality Systems on Multiprocessor Platforms, DATE 14

Method 2+ (M2)



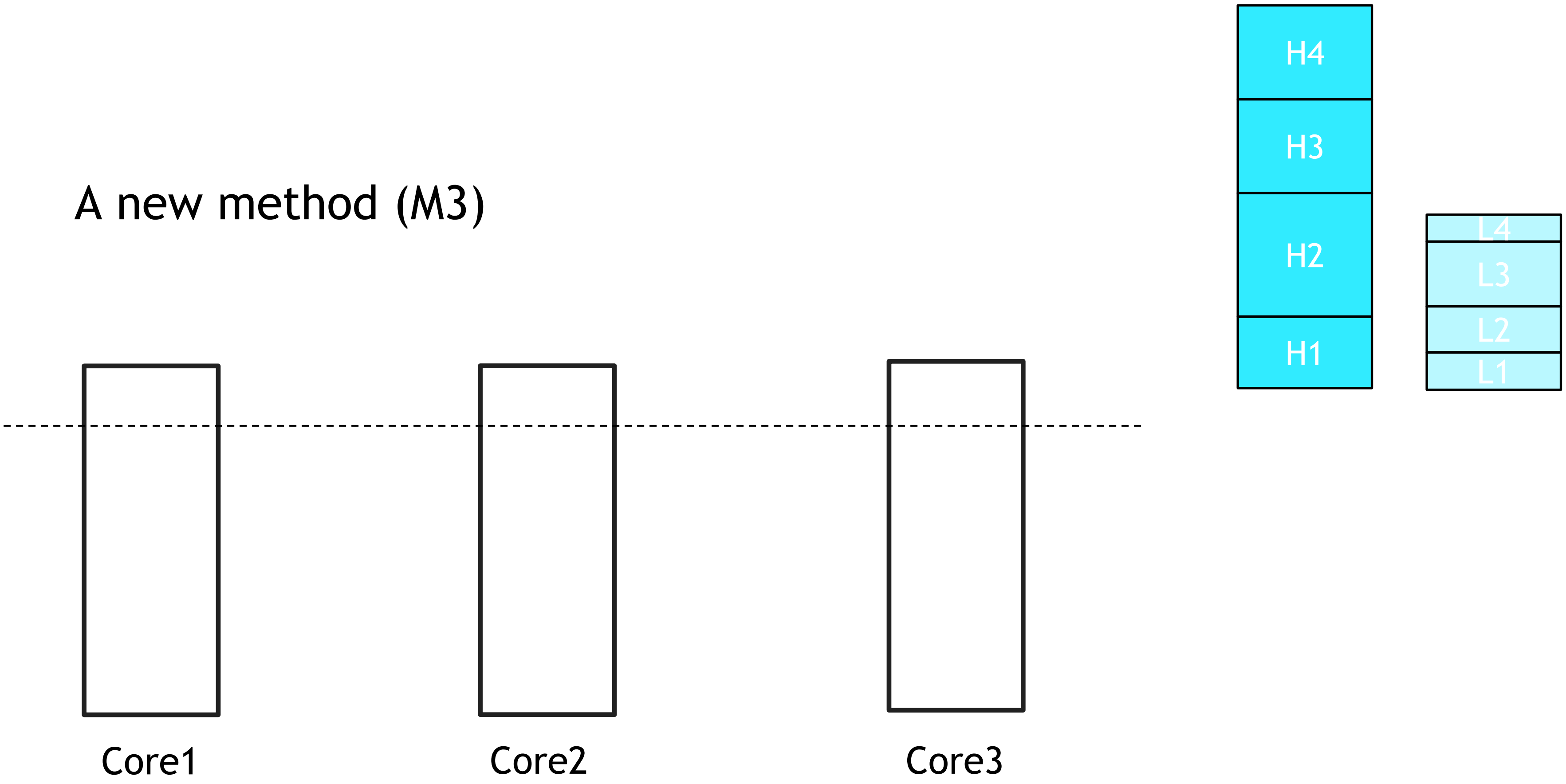
+Gu et al. Partitioned Scheduling of Sporadic Mixed-Criticality Systems on Multiprocessor Platforms, DATE 14

Method 2[+] (M2)

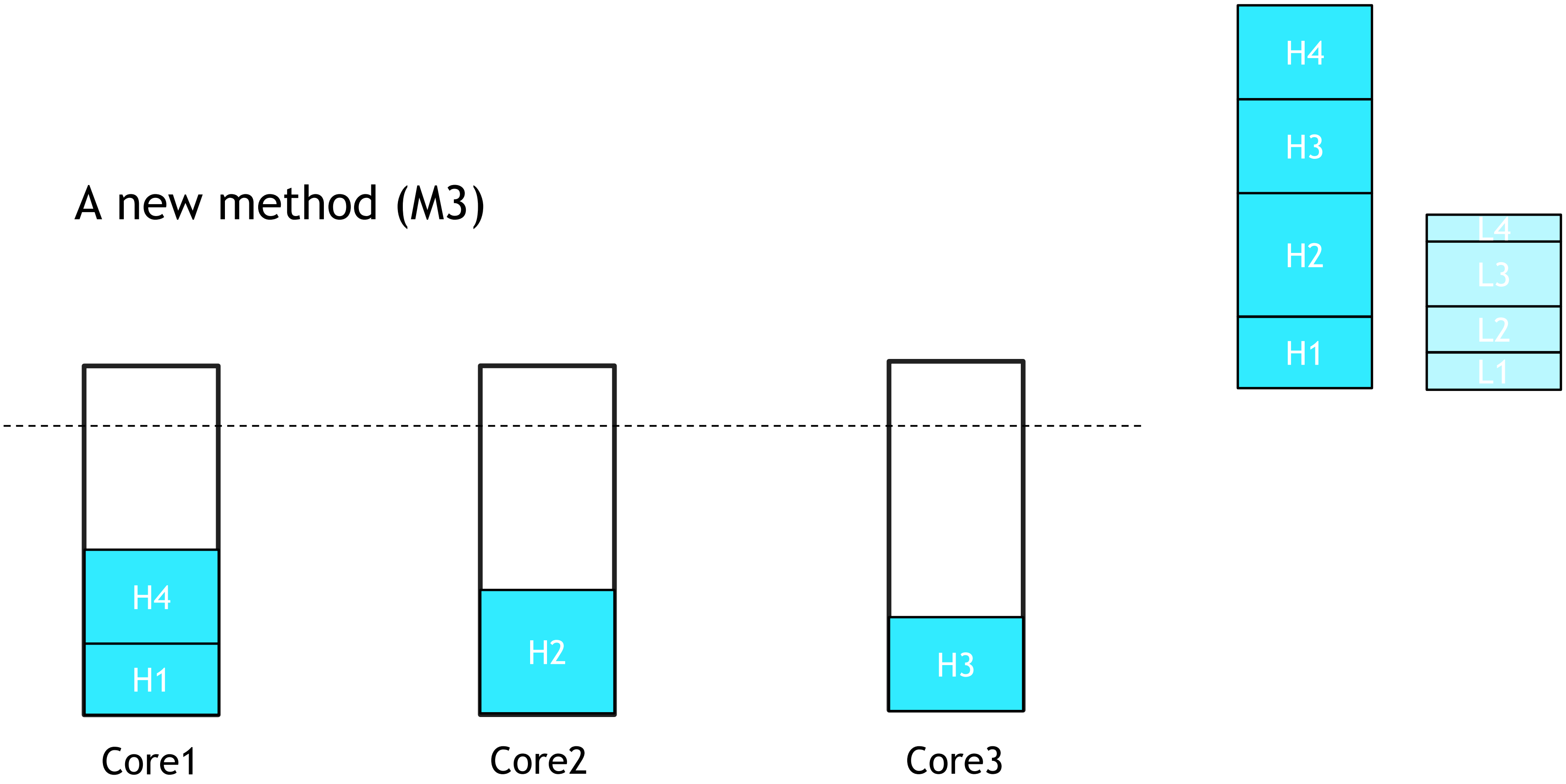


[+]Gu et al. Partitioned Scheduling of Sporadic Mixed-Criticality Systems on Multiprocessor Platforms, DATE 14

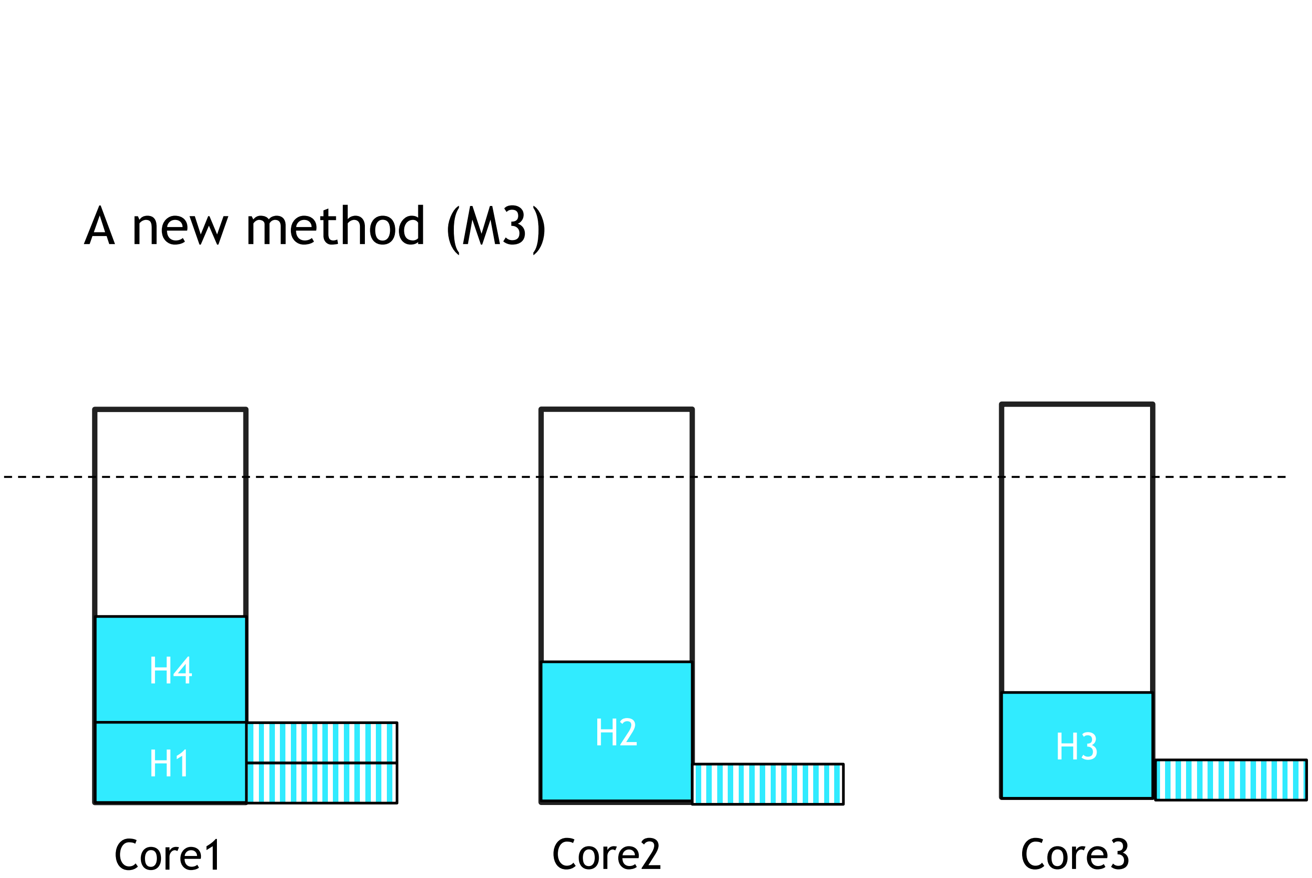
A new method (M3)



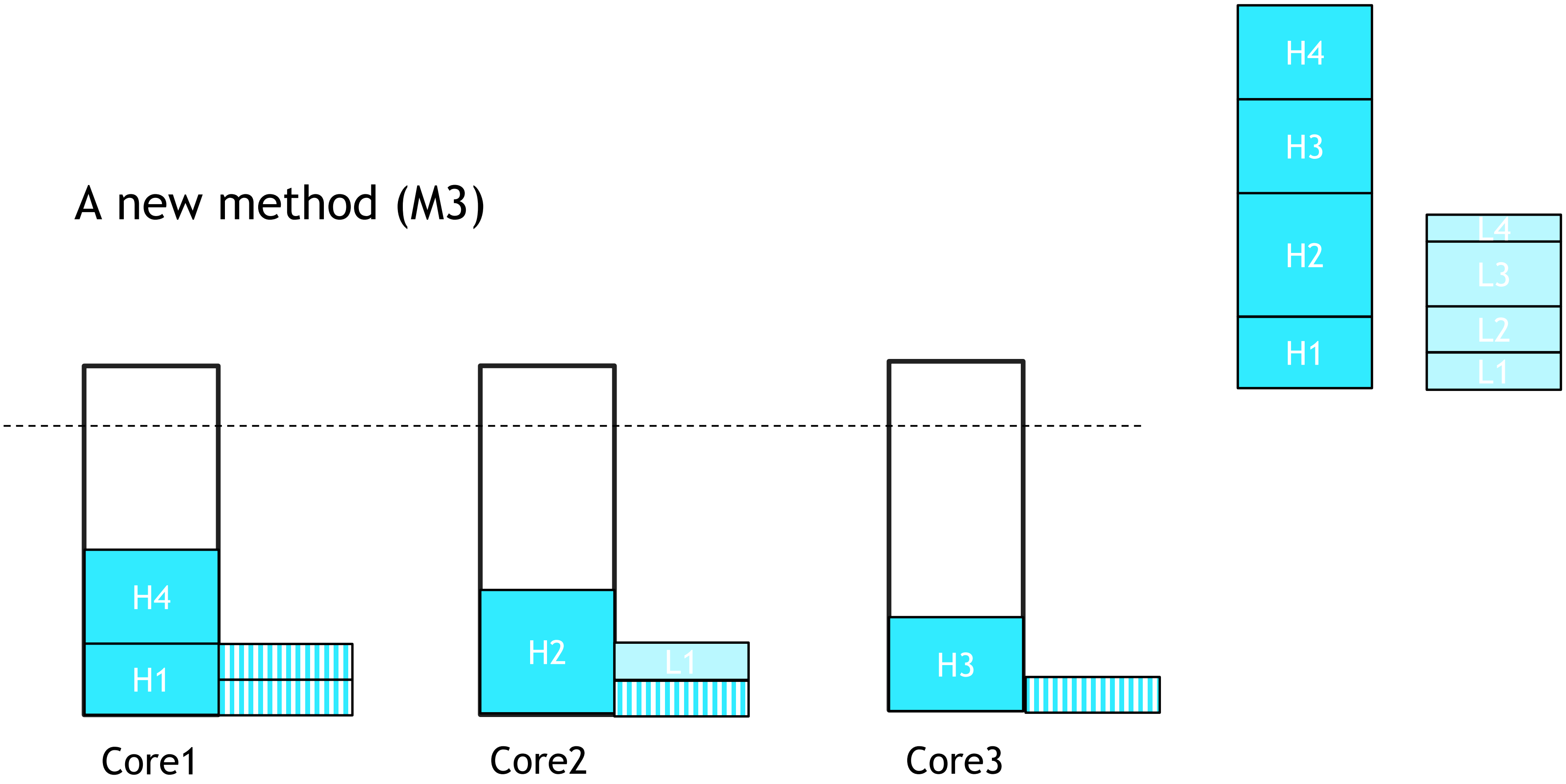
A new method (M3)



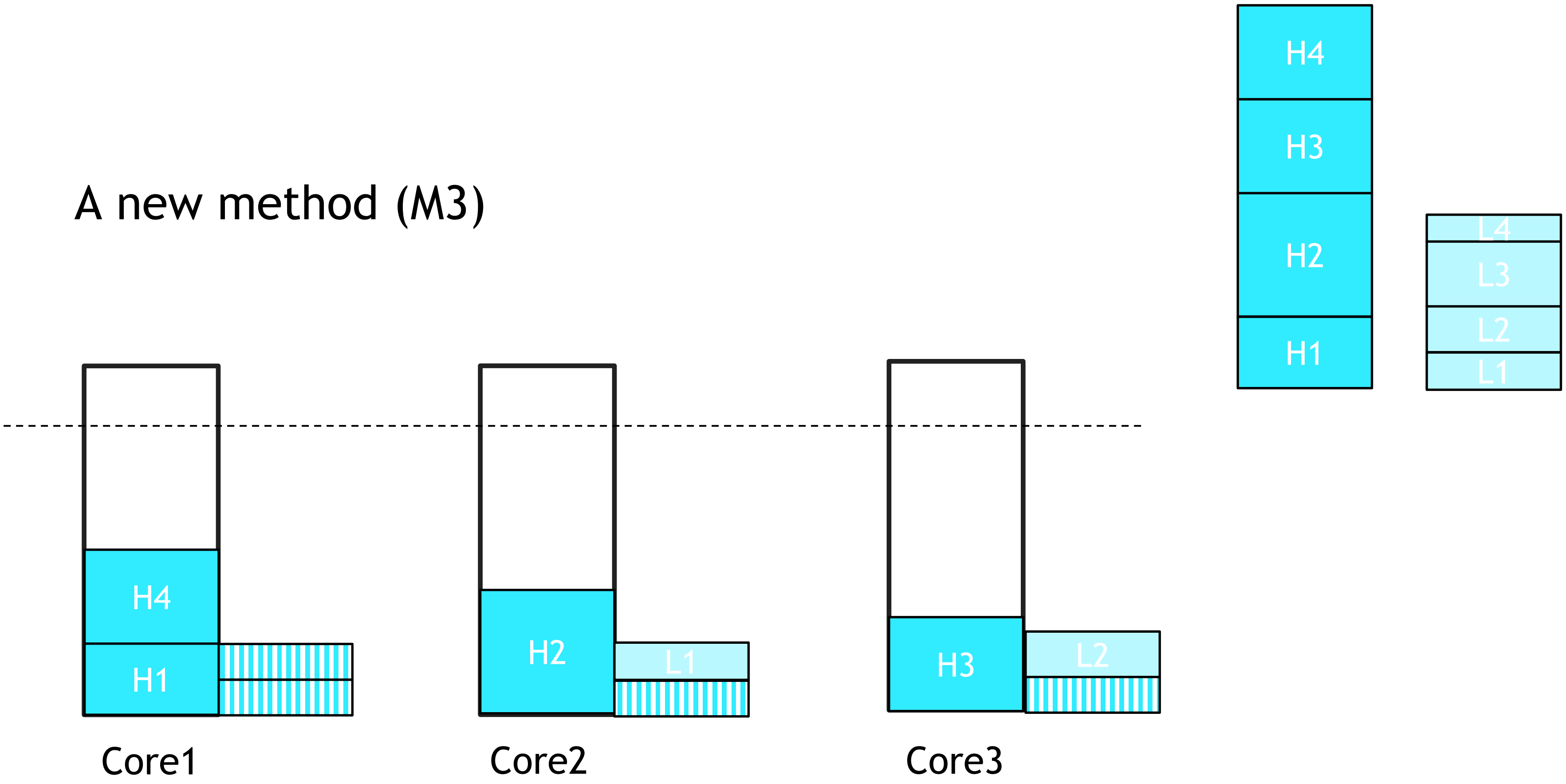
A new method (M3)



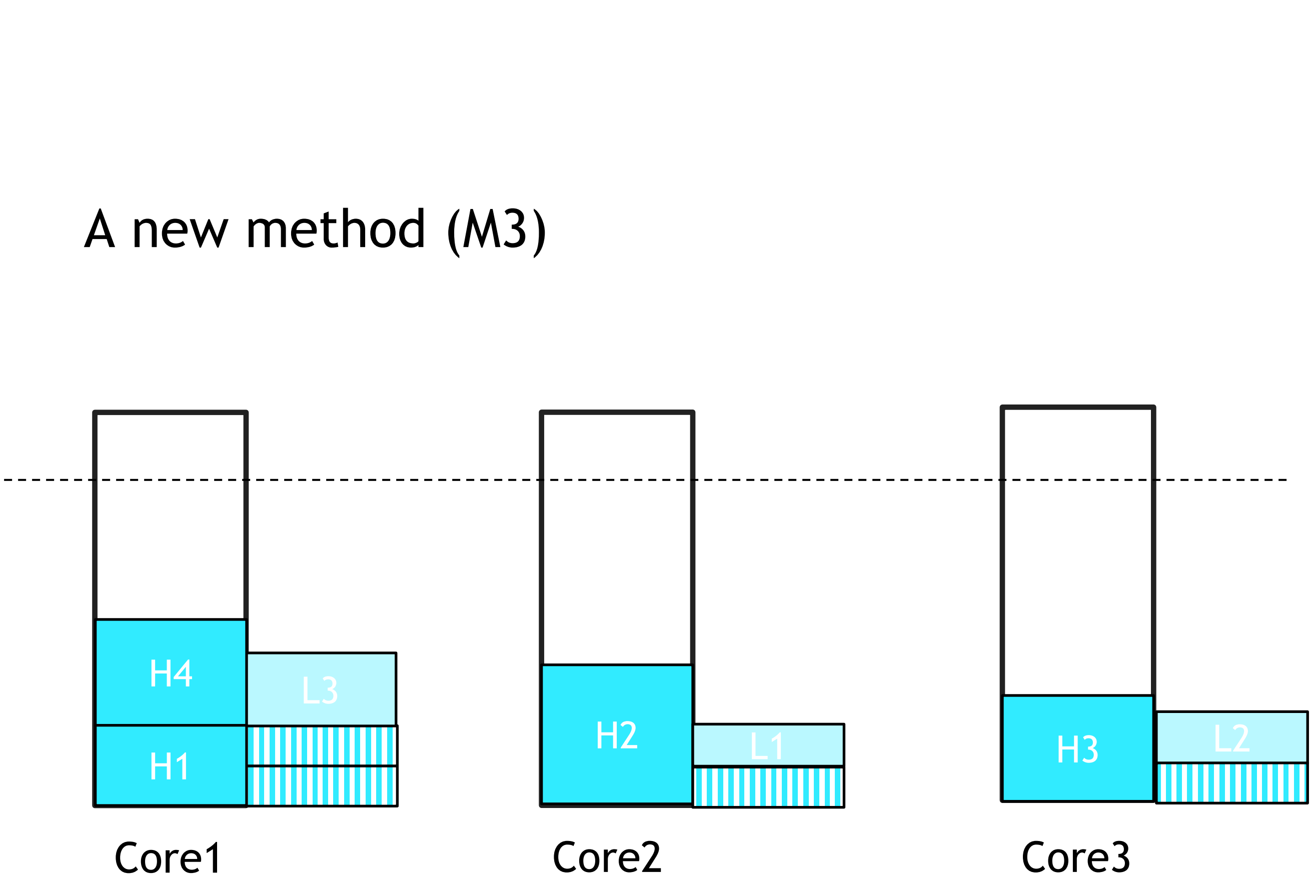
A new method (M3)



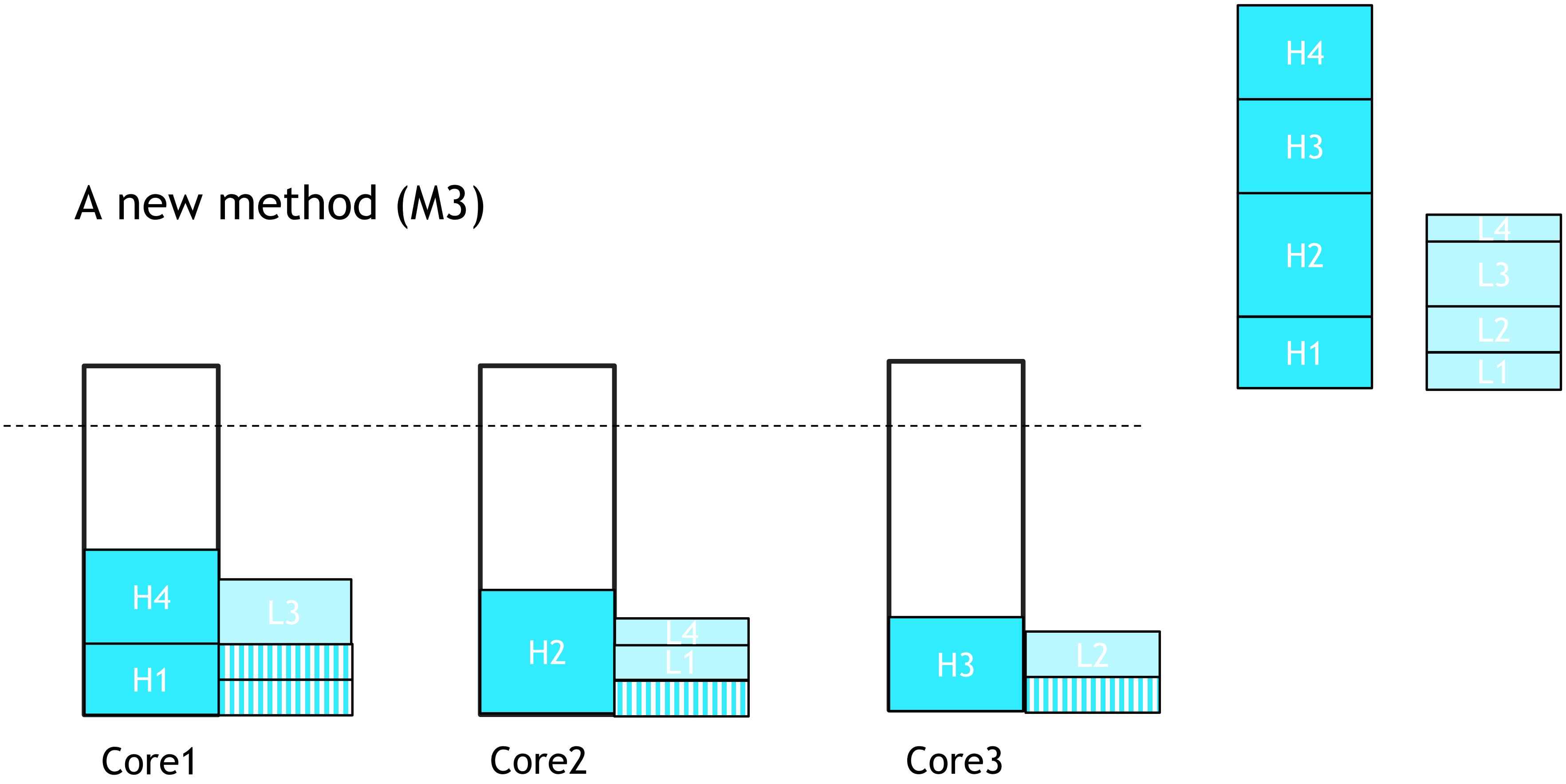
A new method (M3)



A new method (M3)



A new method (M3)



Is that all?

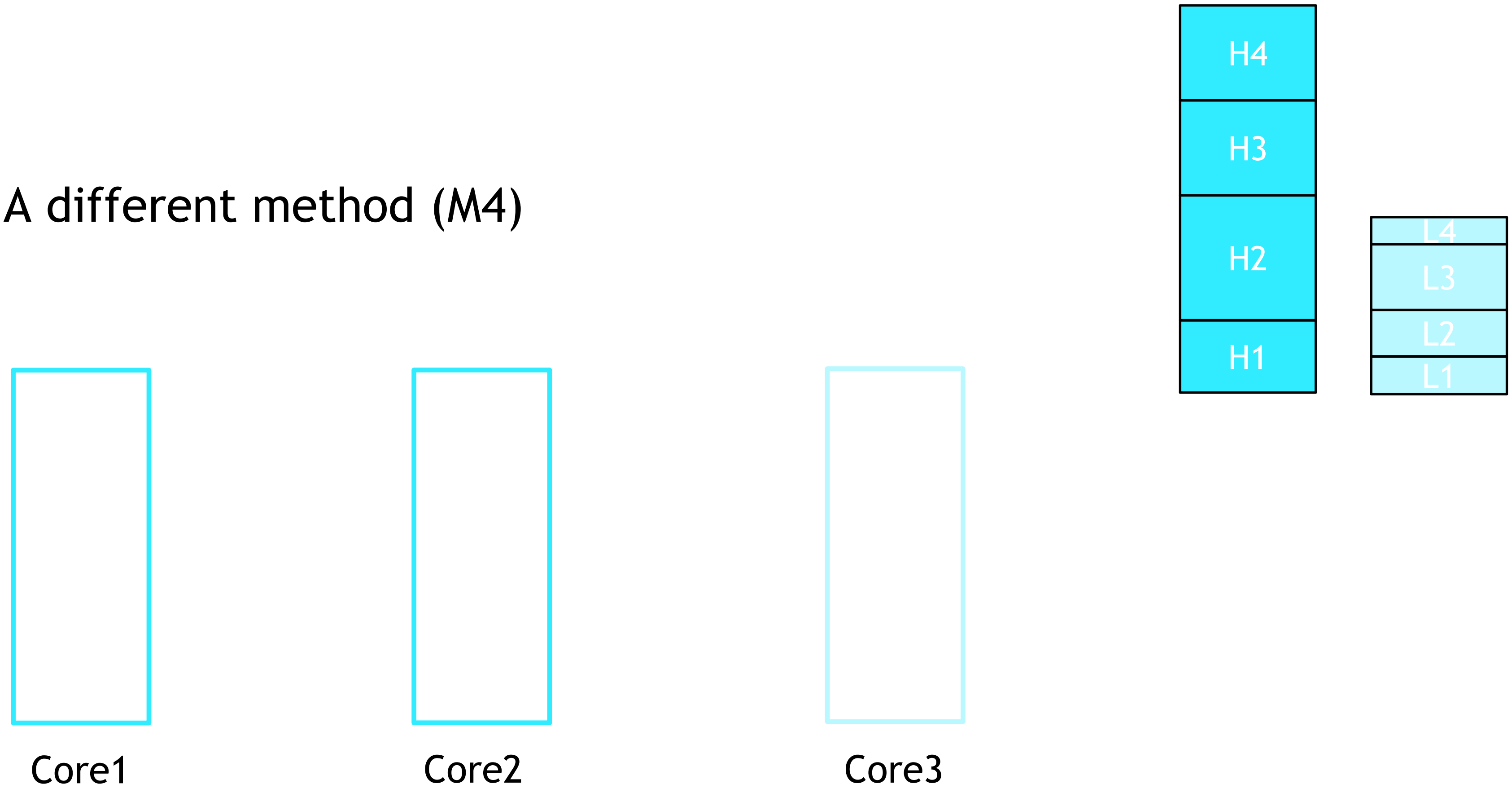
MC scheduling → mix tasks to reduce # cores

Is that all?

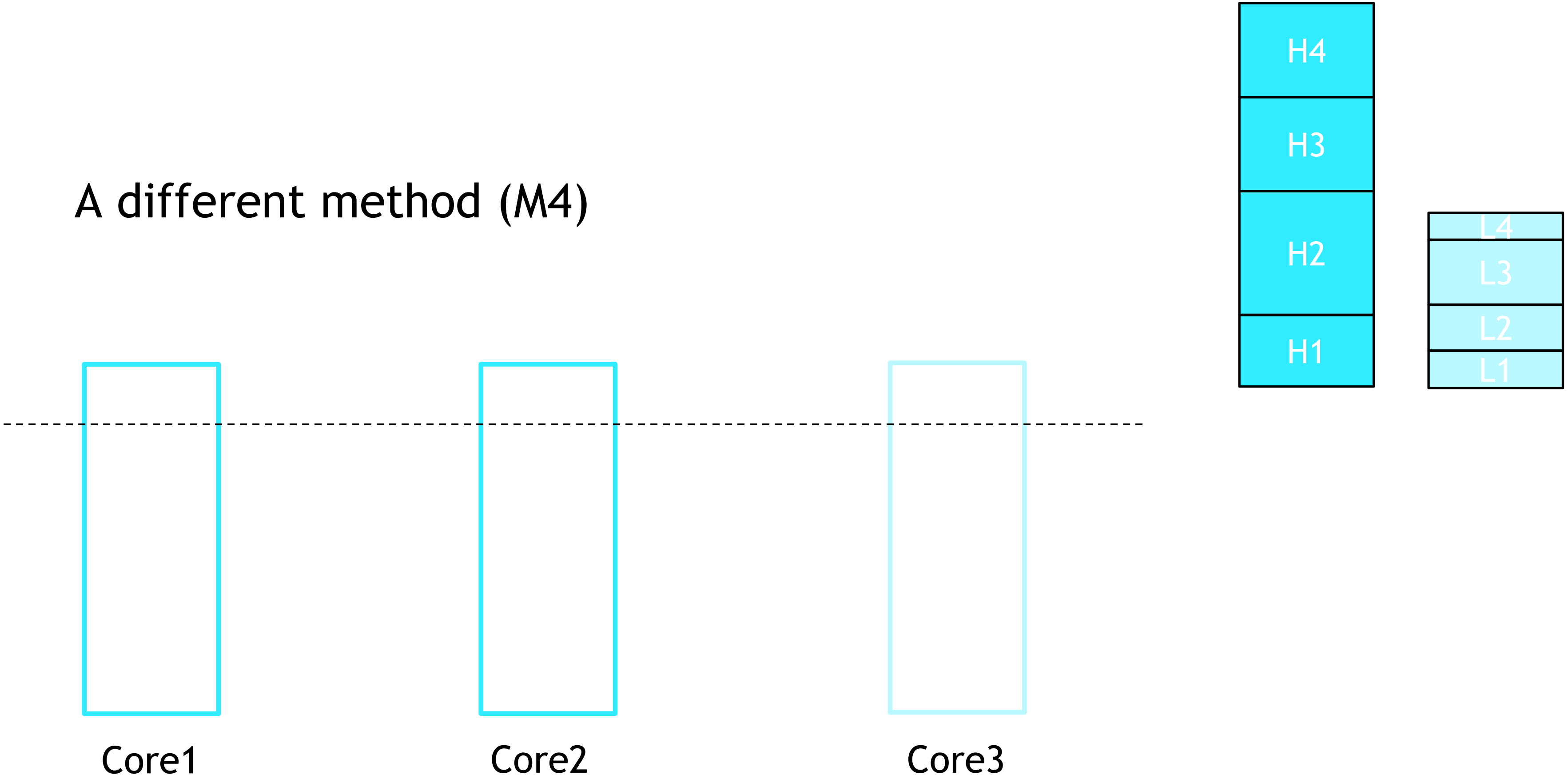
MC scheduling → mix tasks to reduce # cores

If we are conserving energy, would mixing still be good?

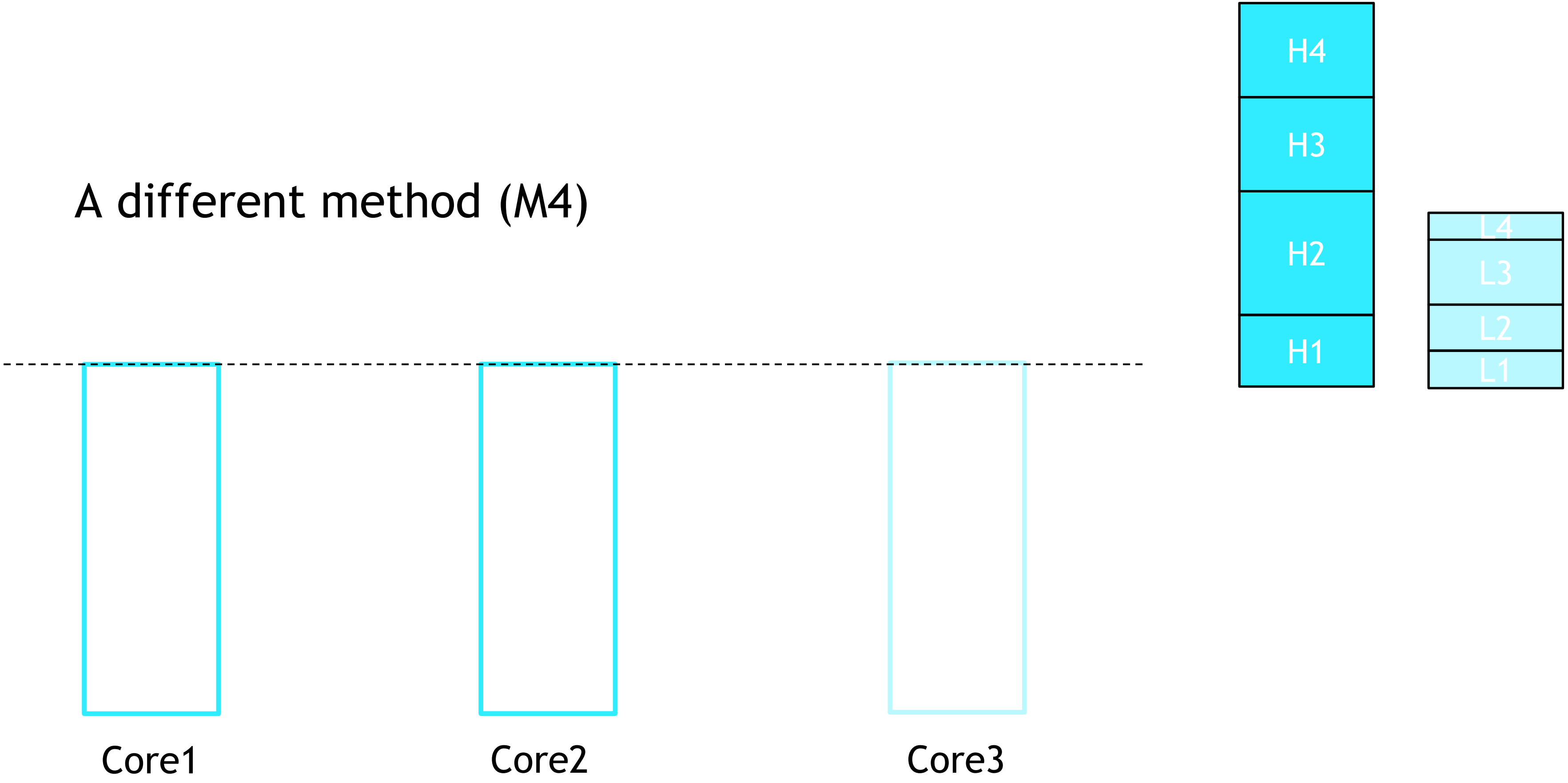
A different method (M4)



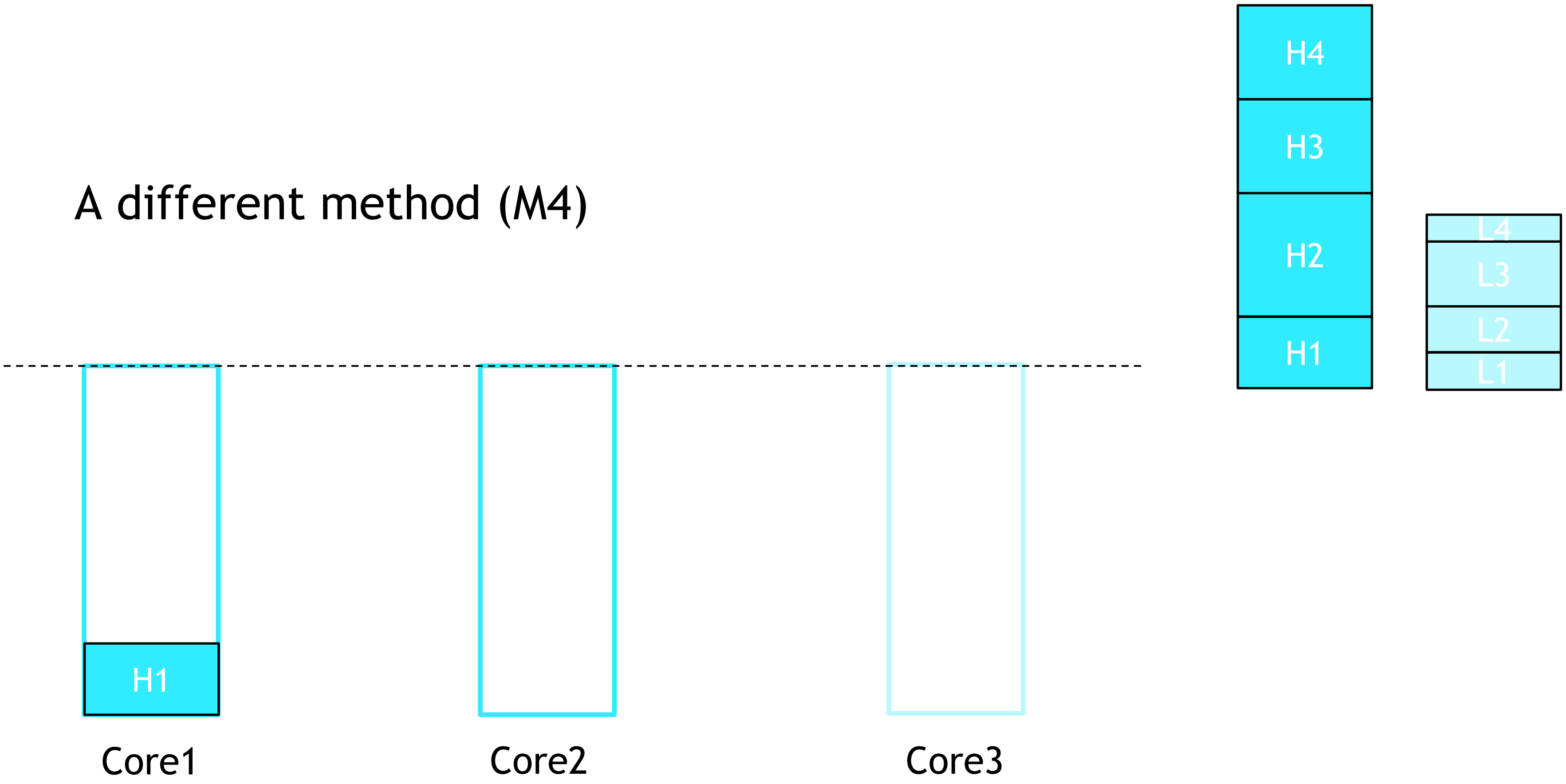
A different method (M4)



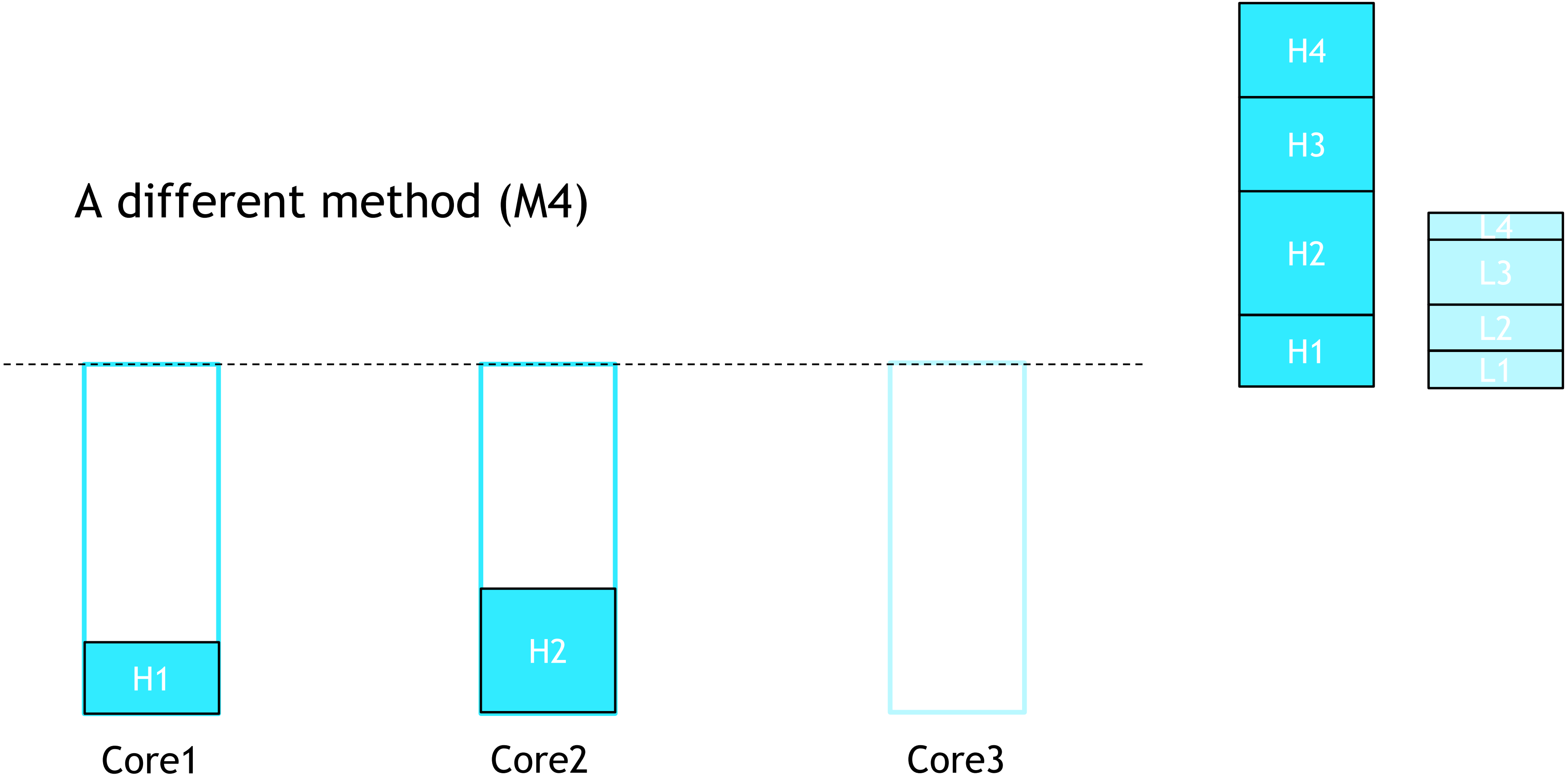
A different method (M4)



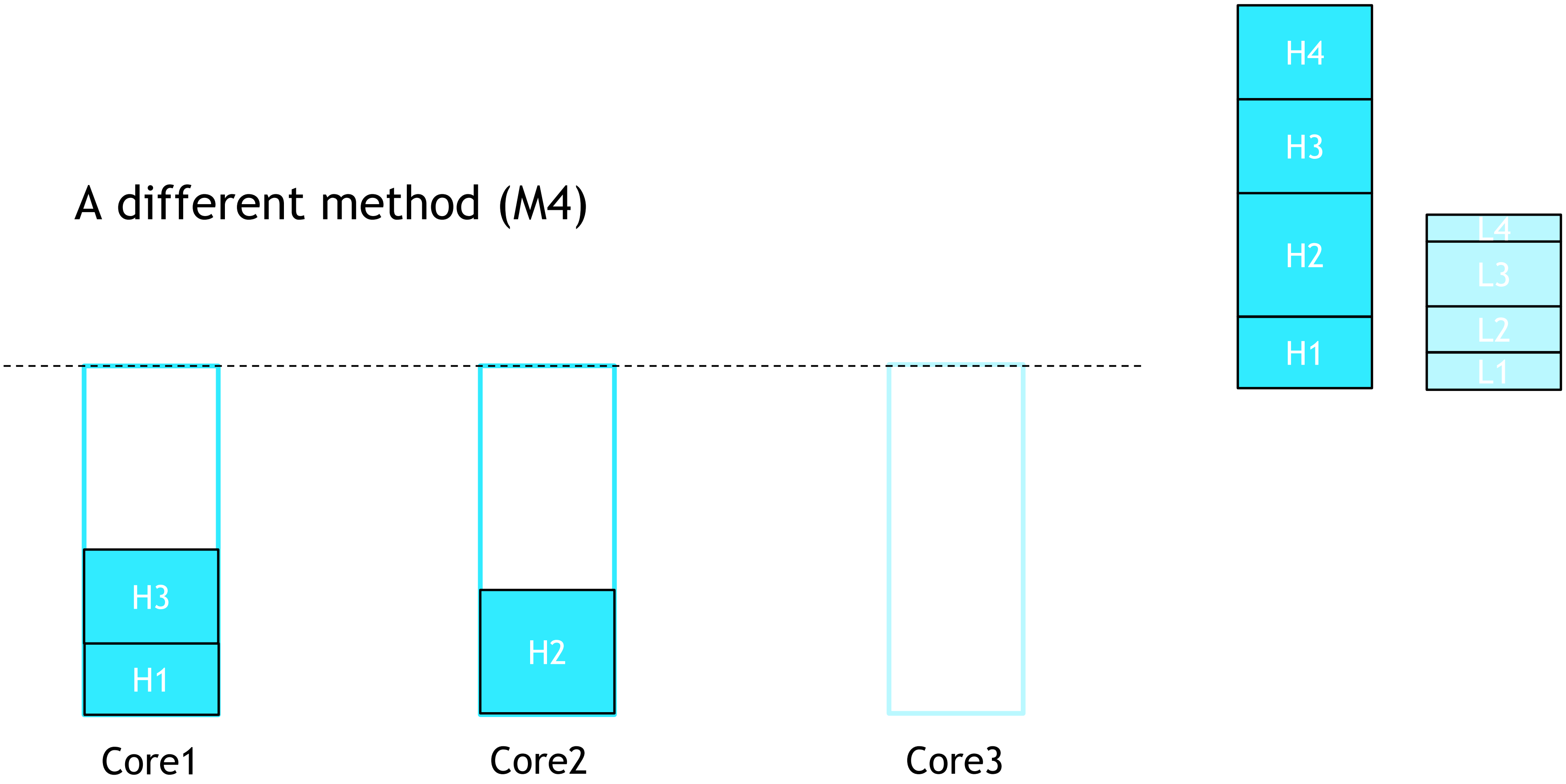
A different method (M4)



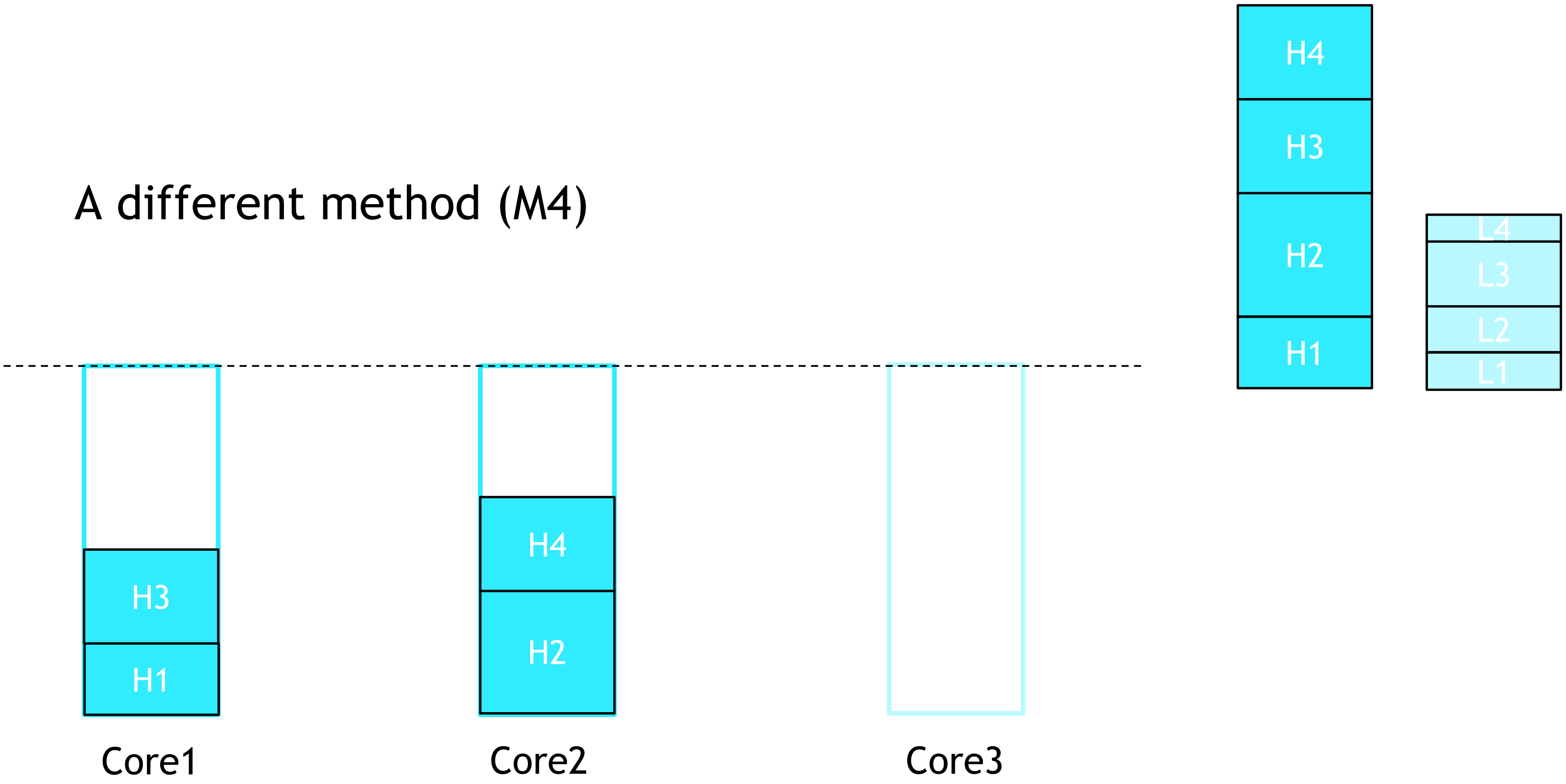
A different method (M4)



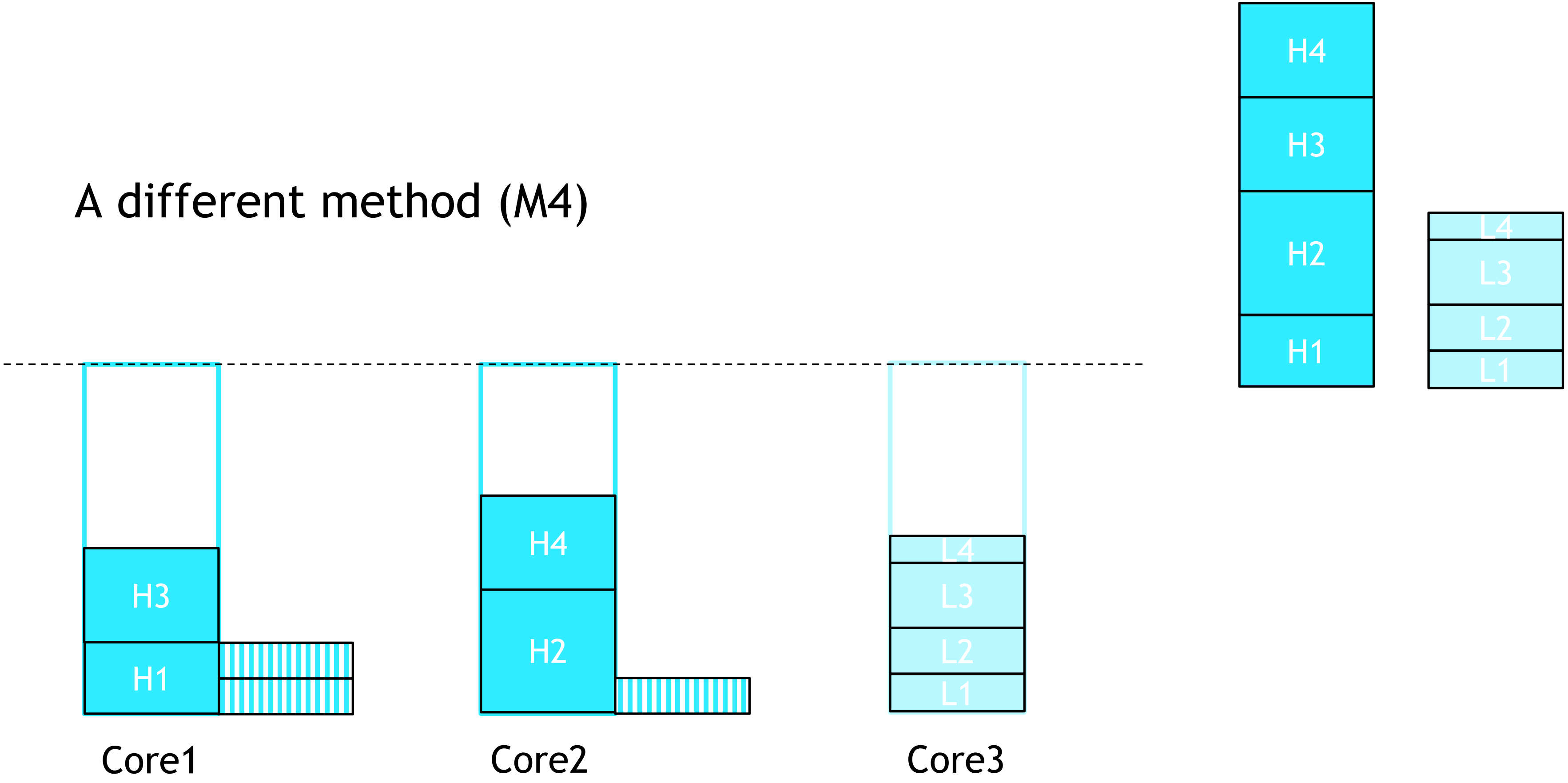
A different method (M4)



A different method (M4)



A different method (M4)



How good are the proposed solutions?

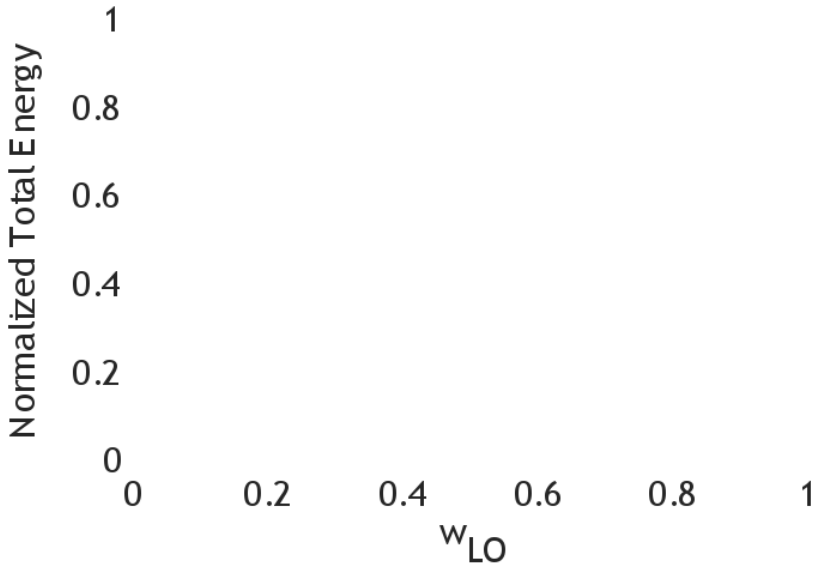
Flight management system – 7 HI, 4 LO

$$\alpha = 2, \beta = \frac{1.76W}{GHz^{\alpha}}, f_{min}/f_b/f_{max} = 0.5/0.8/1\ GHz, P_s = 0.8W$$

Random simulations – 1000 task-sets per utilization

$$\alpha = 2, \beta = \frac{1.76W}{GHz^{\alpha}}, f_{min}/f_b/f_{max} = 0.5/0.8/1\ GHz, P_s = 0.5W$$

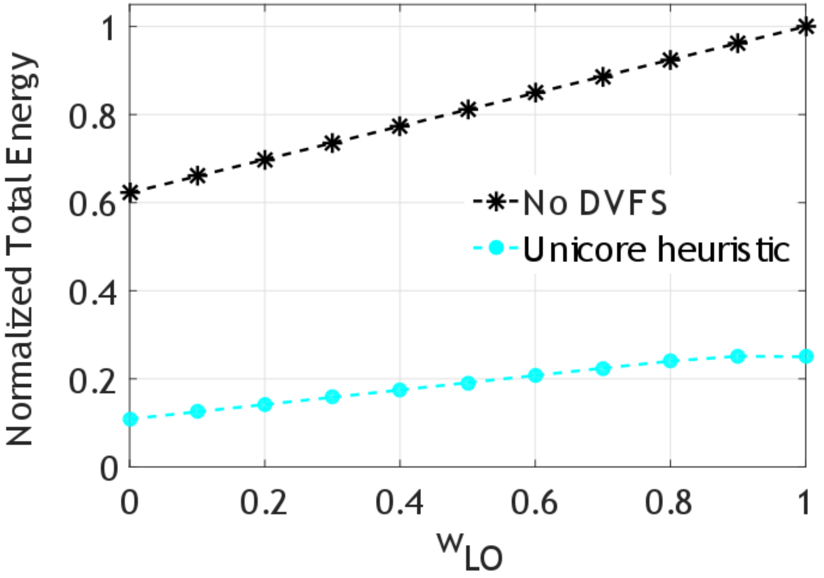
FMS



FMS

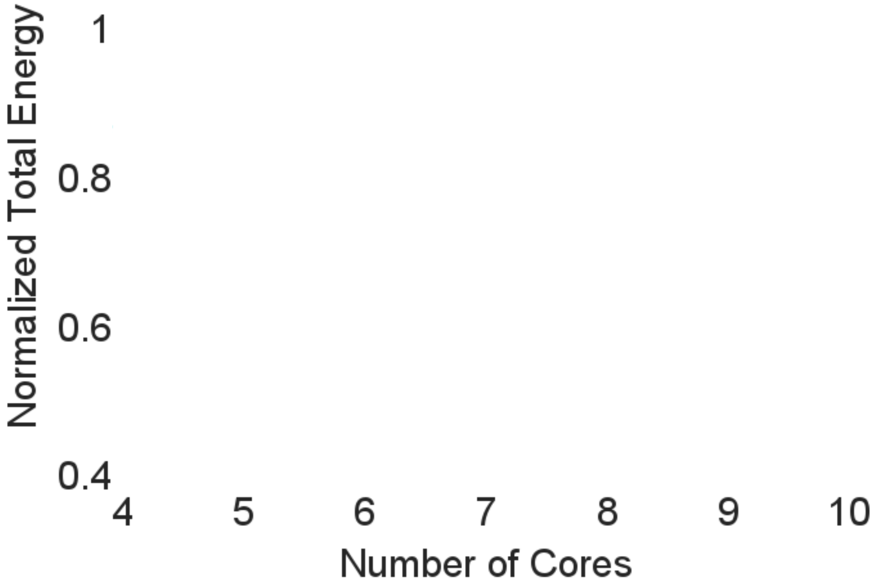
30%~50% energy reduction

Contribution factor does impact!



Random simulations

$U = 3, w_{LO} = w_{HI} = 0.5$



Random simulations

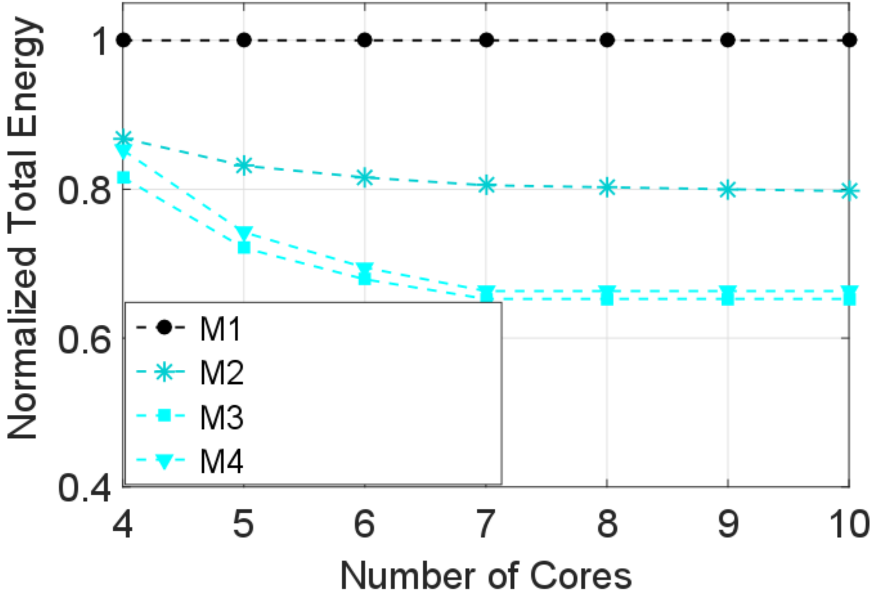
$U = 3, w_{LO} = w_{HI} = 0.5$

Proposed methods perform well

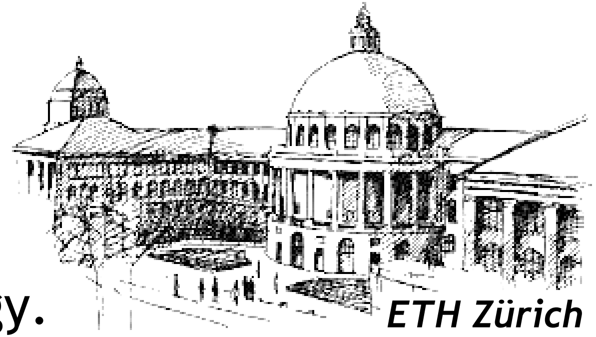
$7 cores\!: \sim 30\% \; (\sim 20\%) \; more \; saving \; than \; M1 \; (M2)$

Isolation pays-off!

$very \; close \; to \; M3$



Minimize static & dynamic energy.

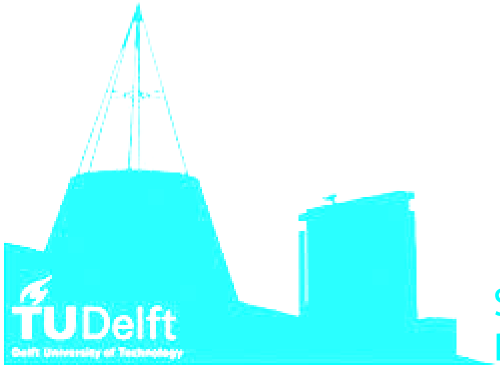


ETH Zürich

Pengcheng Huang
Georgia Giannopoulou
Lothar Thiele

Optimal & fast numerical solutions for unicore.

Energy-aware mappings for multicore.



Sujay Narayana
R.Venkatesha Prasad

If energy is a concern, isolate!

ETH Zürich

Pengcheng Huang
Georgia Giannopoulou
Lothar Thiele

TU Delft
Delft University of Technology

Sujay Narayana
R.Venkatesha Prasad