A Unified Online Directed Acyclic Graph Flow Manager for Multicore Schedulers

Karim Kanoun (EPFL), David Atienza (EPFL),
Nicholas Mastronarde (UB) and Mihaela van der Schaar (UCLA)

Karim.Kanoun@epfl.ch
Embedded Systems Laboratory
EPFL, Switzerland

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

University at Buffalo
The State University of New York

UCLA

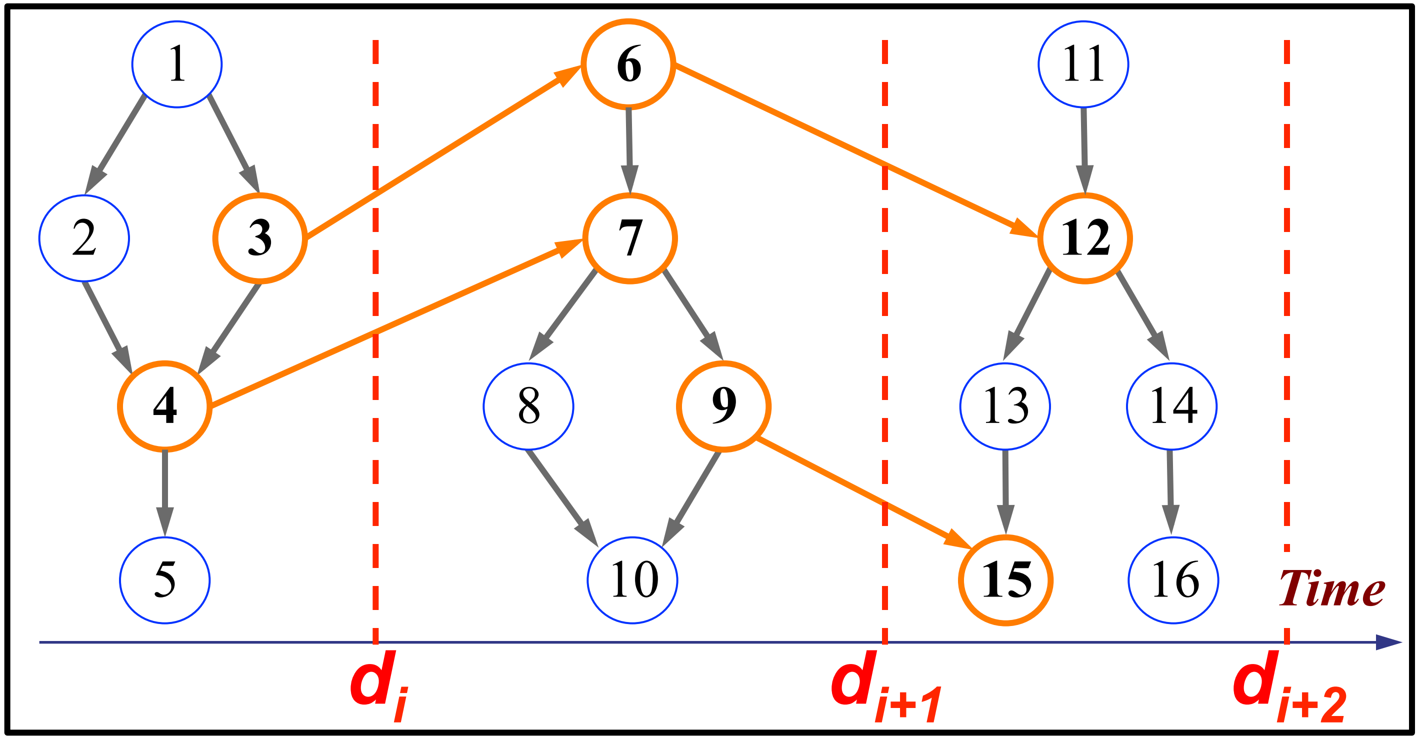
ASP-DAC 2014

Outline

- Motivation
- Problem statement
- Proposed solution: Online DAG Flow Manager
- Experimental setup and results
- Conclusion

Modeling a task-graph application with a general DAG model

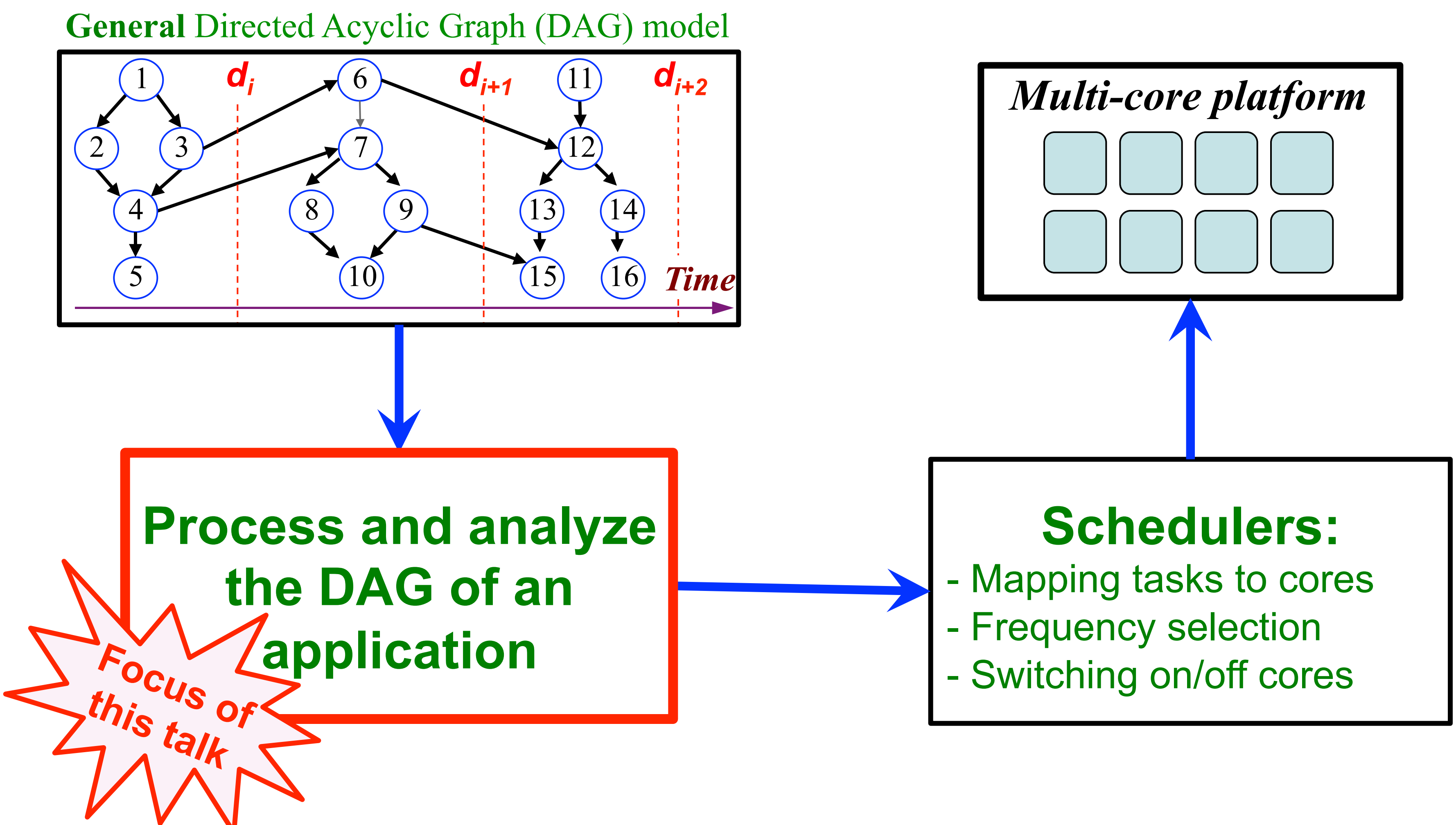
General Directed Acyclic Graph (DAG) model



Dependency inside a deadline tasks set

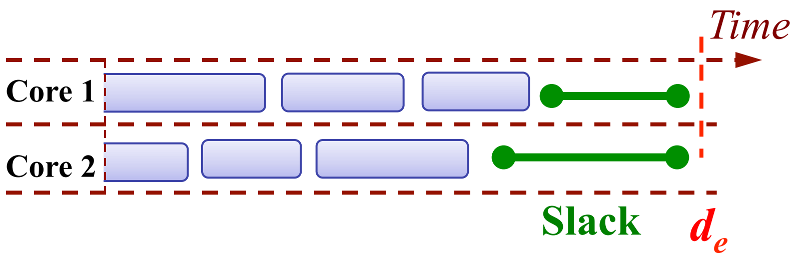
Dependency between 2 deadlines tasks set

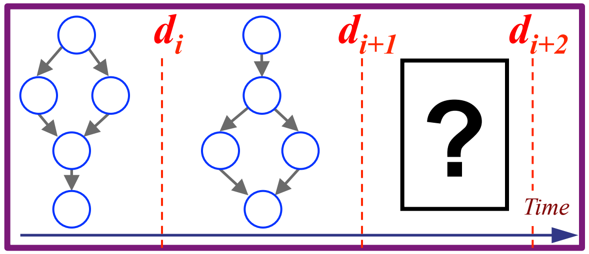
From the application layer to the hardware layer



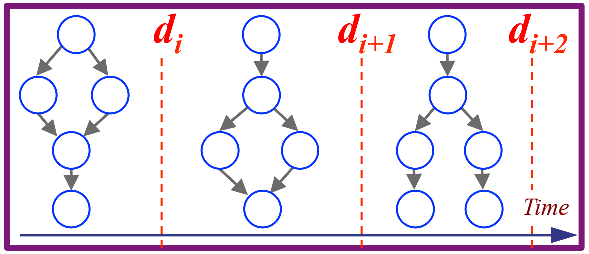
Limitation of existing DAG analysis solutions

- Static solutions
  - Unsuitable for applications with non-deterministic workload
  - Not applicable when the DAG model is determined at run-time

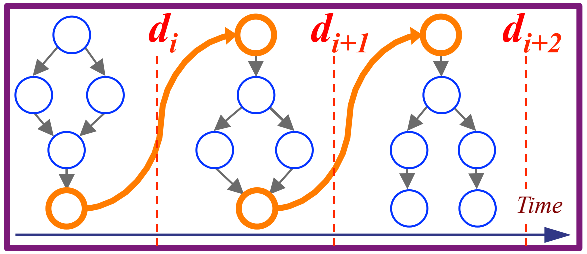




- Online solutions
  - Designed for their own specific schedulers (most of the time)
  - Limitation on the DAG model



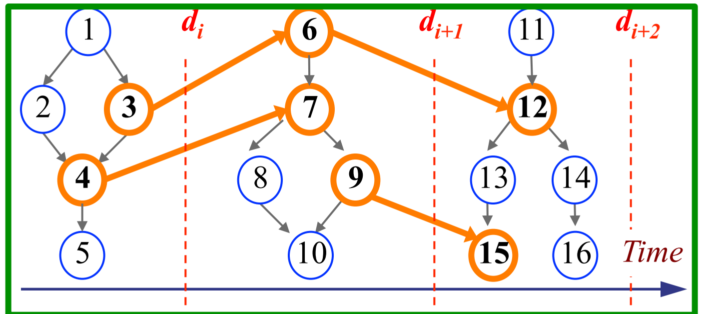
Independent deadlines



Dependent deadlines
(only fork join)

Problem statement

General DAG model

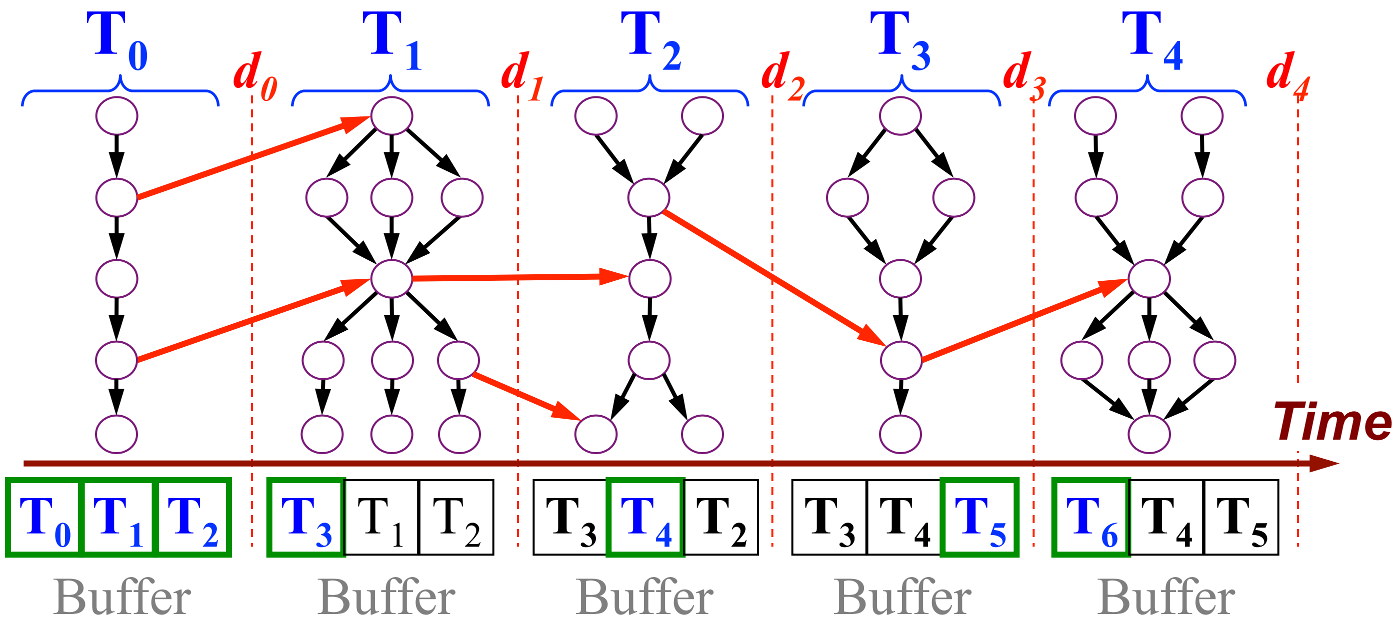


- Unified DAG analysis solution to assist schedulers:
  - Online, low-complexity and scheduler-independent
  - Process all possible applications (general DAG)
  - Provide detailed tasks dependencies to schedulers

Any external scheduler

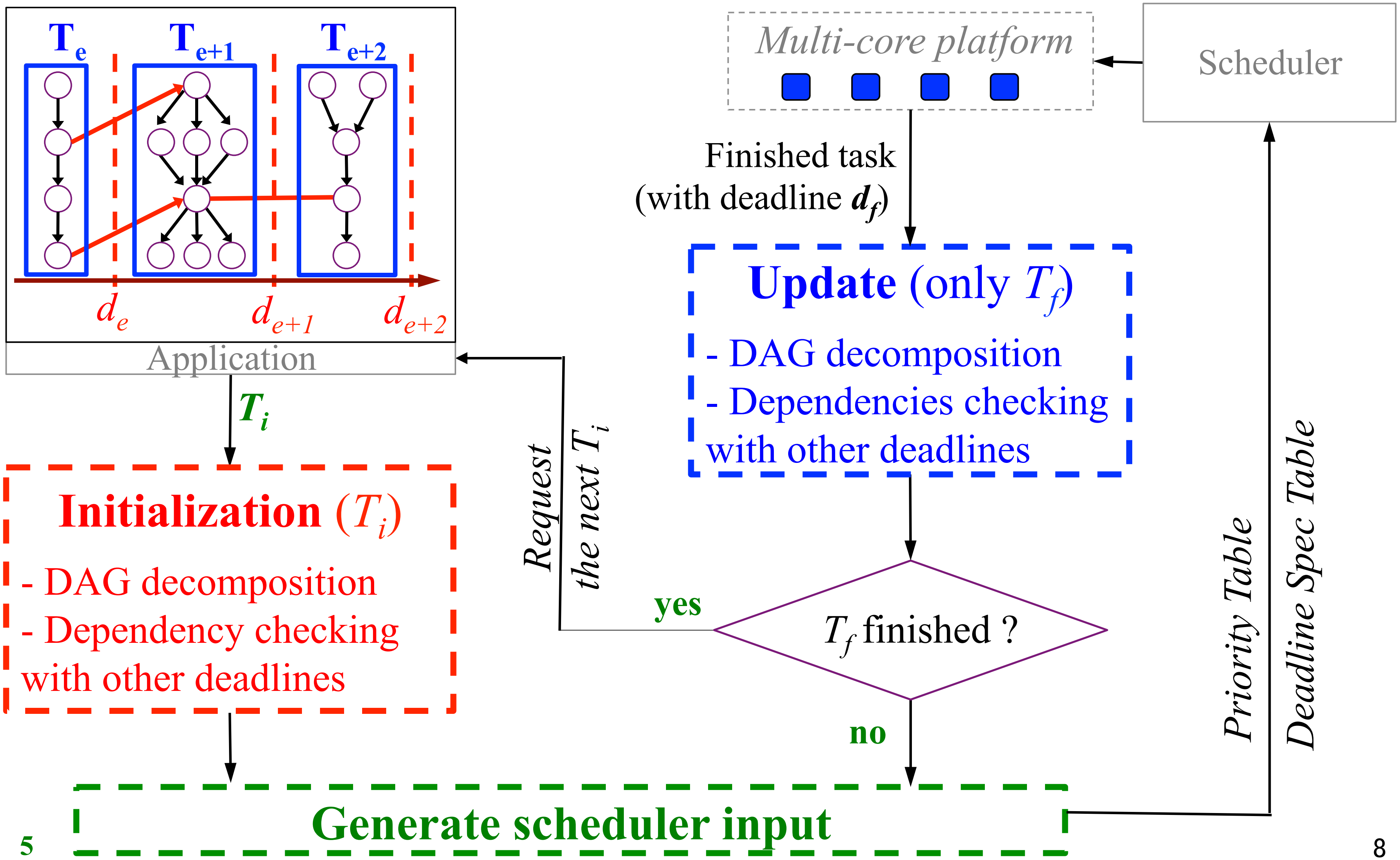
DAG Flow Manager (DFM) Input and key point

- Using a look-ahead window buffer to process the full DAG
- Dependencies in red color are managed with a separated list



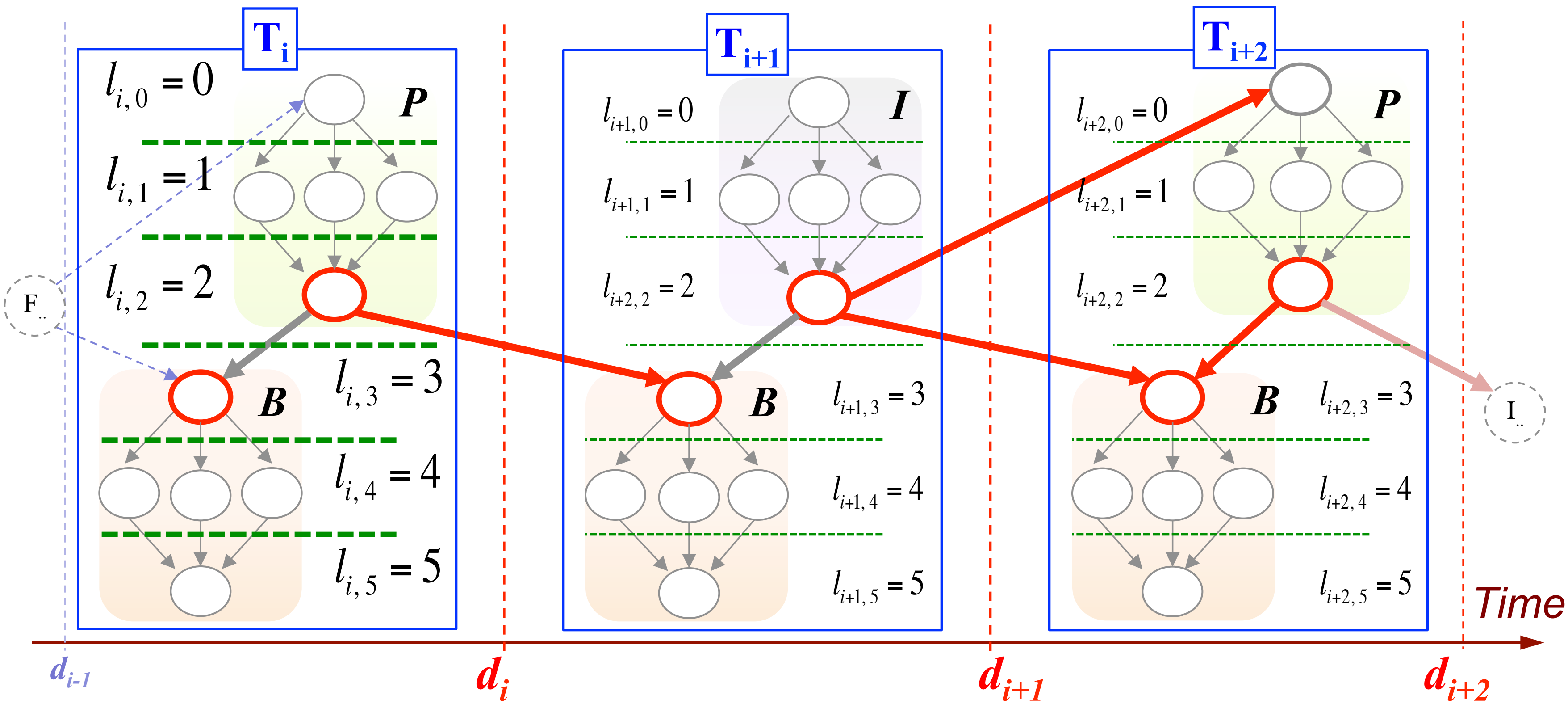
- DFM Input (for each T_i)
  - Adjacency matrix
  - Deadline value
  - List of edges connecting T_i with T_{i+l} (with $l \neq 0$)

DAG Flow Manager (DFM): Overview



DFM
Tasks sets decomposition

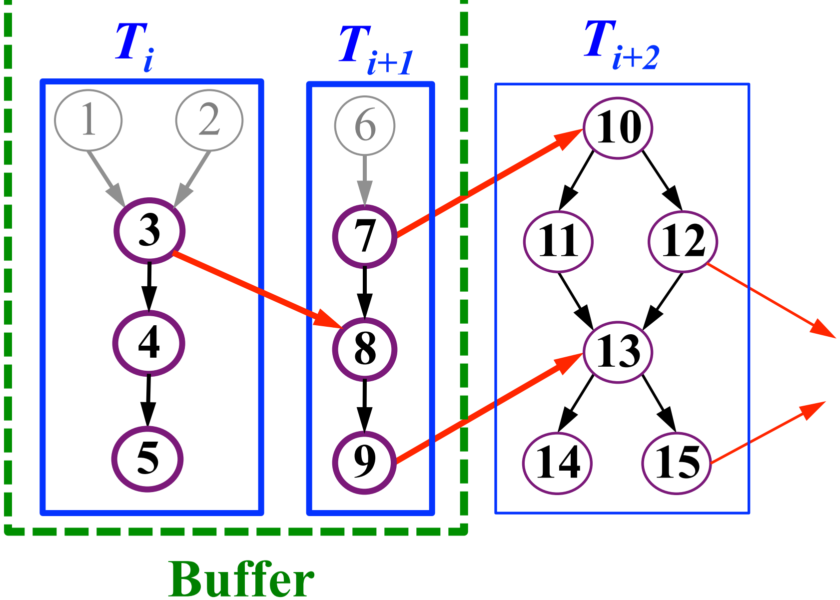
- The decomposition is applied during the **initialization** and the **update** phases
  - Example of an H.264 video decoder DAG decomposition



$l_{i,j}$ Depth level	T_j Group of tasks having the same deadline

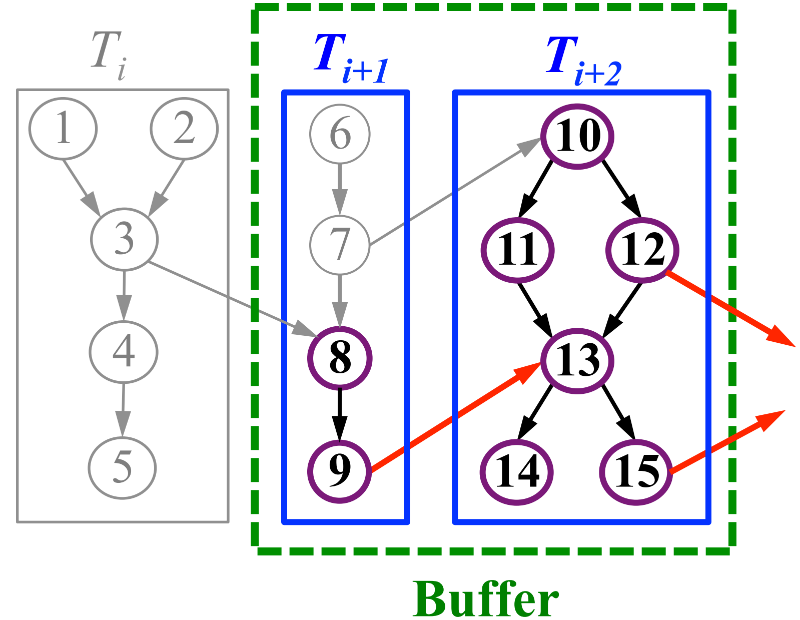
DFM
Managing dependencies between the deadlines

Update phase



- If task 3 finishes then edge [3-8] is cleared normally.
- If task 7 finishes then edge [7-10] is stored in the **list of non-cleared dependencies.**

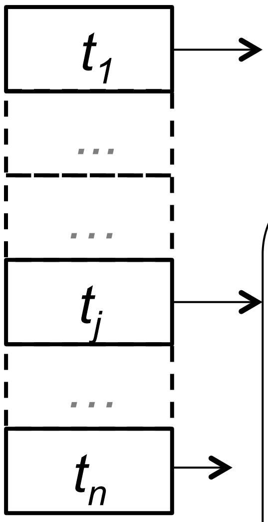
Initialization phase (T_{i+2})



- Edge [7-10] is detected in the **list of non-cleared dependencies.** It is then cleared and removed from the list

DFM
Prepared scheduler input: Priority Table

- Tasks in the Priority Table are sorted according to their:
  - (1) Deadline
  - (2) Depth level in T_i
  - (3) Estimated workload



Priority Table

allow detecting if a task is ready to be scheduled

Scheduler input for each task

- The total number of parent tasks that it still depends on

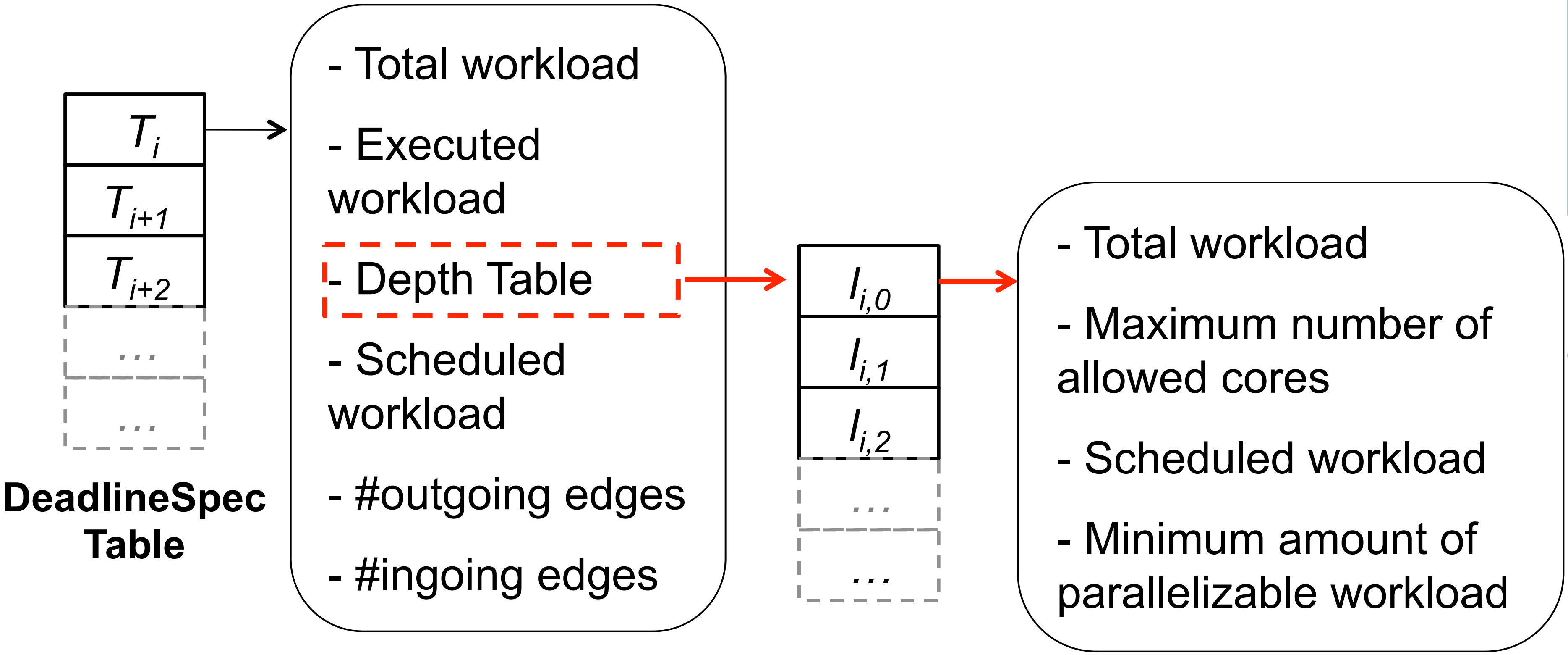
- Workload

- Earliest release time

- Critical path workload to its deadline

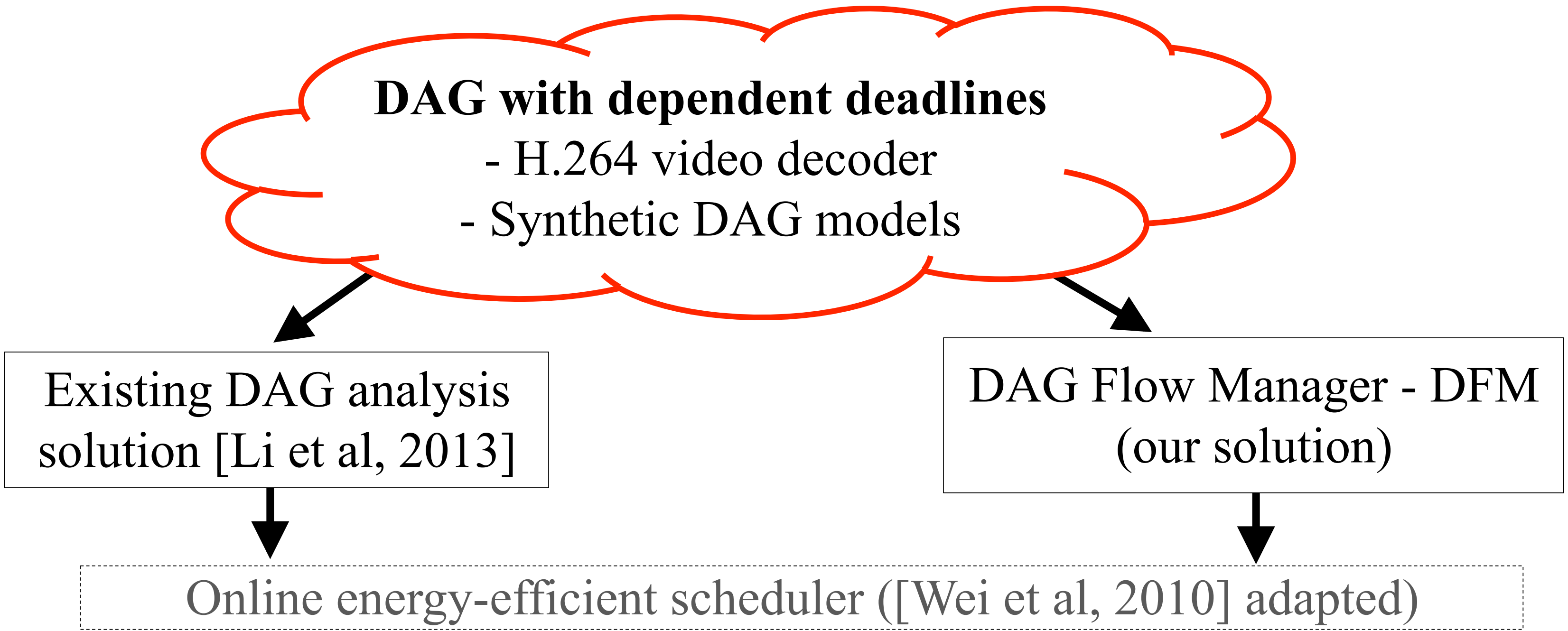
DFM
Prepared scheduler input: DeadlineSpec Table

- T_i refers to the tasks set with deadline d_i
- The DeadlineSpec Table is used to track the overall progress of each tasks set T_i.



Experimental Setup

- H.264 video decoder tasks workload measured with MPARM simulator [*Benini, et al, 2005*] and using power figures of *90nm* technology node
- Synthetic DAGs generated with GGEN tool [*Cordeiro, et al, 2010*]



[Benini, et al., 2005] *L. Benini, et al., "Mparm: Exploring the multi-processor soc design space with systemc," J. VLSI Signal Process. Syst., vol. 41, no. 2, pp. 169–182, Sept. 2005.*

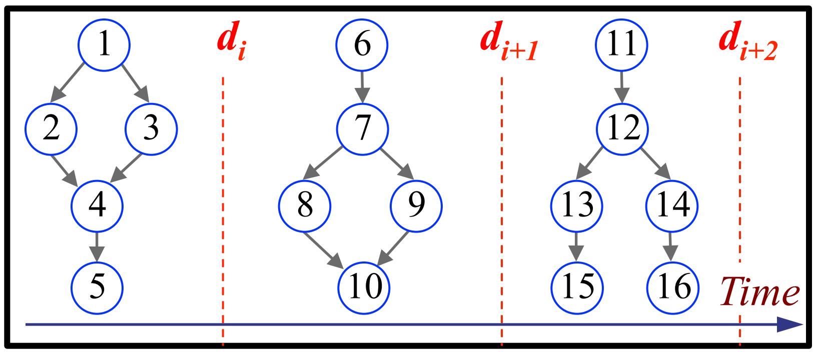
[Cordeiro, et al., 2010] *D. Cordeiro, et al., "Random graph generation for scheduling simulations," in Proc. SIMUTools, 2010.*

[Li, et al., 2013] *J. Li, et al., "Analysis of Global EDF for Parallel Tasks," in Proc. ECRTS, 2013.*

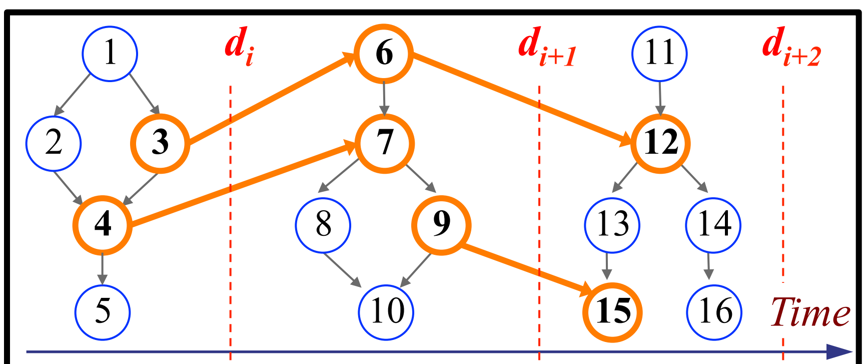
[Wei, et al. 2010] *Y.-H. Wei, et al., "Energy-efficient real-time scheduling of multimedia tasks on multi-core processors," in Proc. ACM SAC, 2010.*

Existing DAG analysis solution

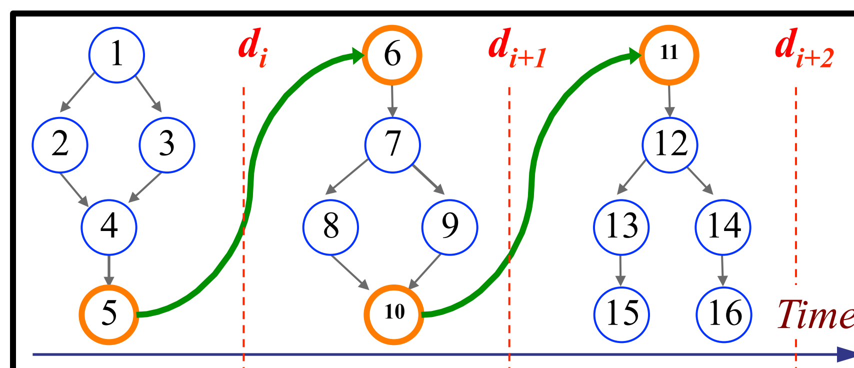
- **[Li et al, 2013]** does not consider dependencies between deadlines tasks sets
  - Forced to convert the general DAG model to the fork-join DAG model and monitor then only one deadline at a time



[Li, et al, 2013] DAG model



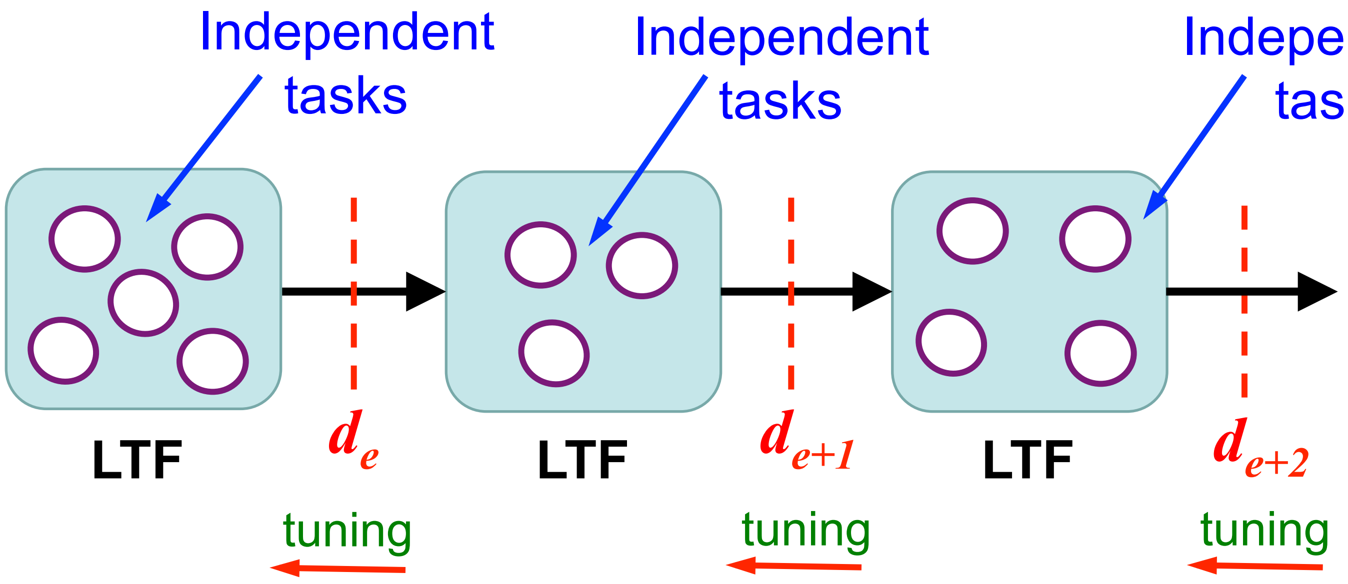
General DAG model



Fork-join DAG model

Connecting an existing scheduler to our DFM Scheduler overview

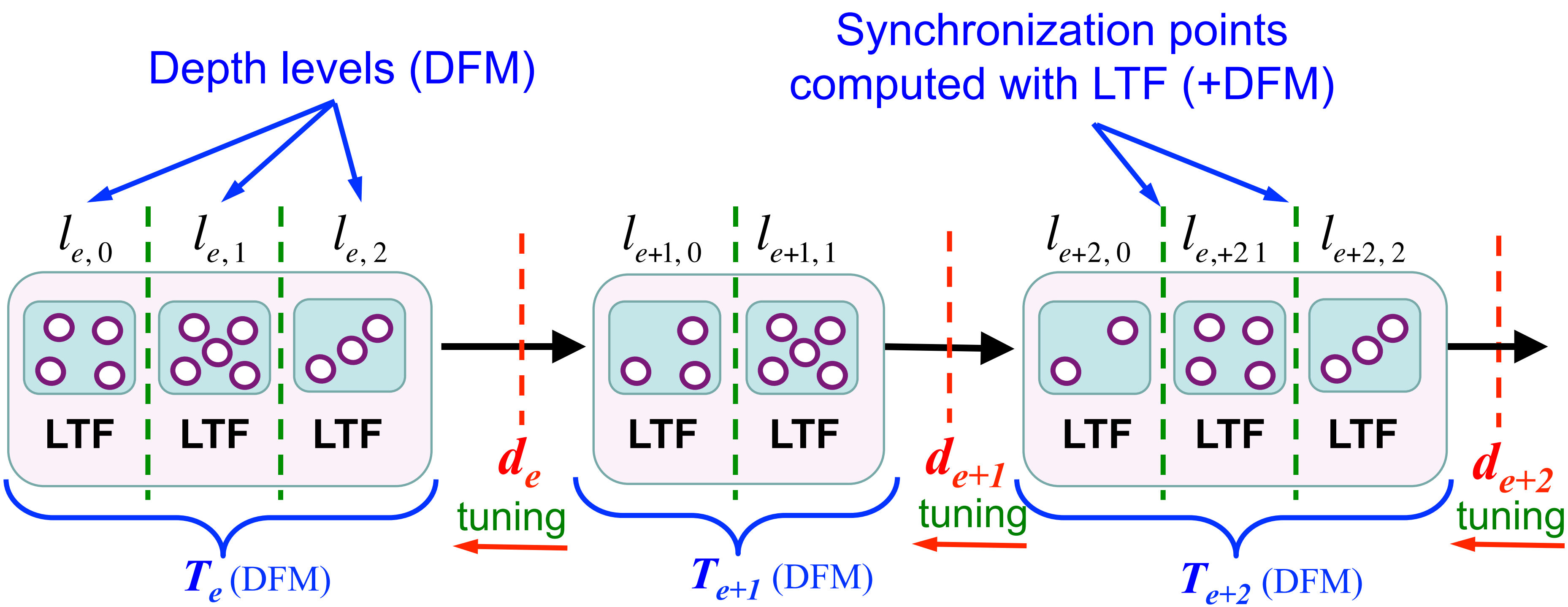
- Existing scheduler *[Wei, et al. 2010] :*
  - Tune the deadline
  - Largest Task First (LTF) on the tasks set with the earliest deadline d_e
  - Set up the minimum frequency based on the LTF and the tuned deadline value



[Wei, et al. 2010] Y.-H. Wei, et al., "Energy-efficient real-time scheduling of multimedia tasks on multi-core processors," in Proc. ACM SAC, 2010.

Connecting an existing scheduler to our DFM
Adaptation to the general DAG model

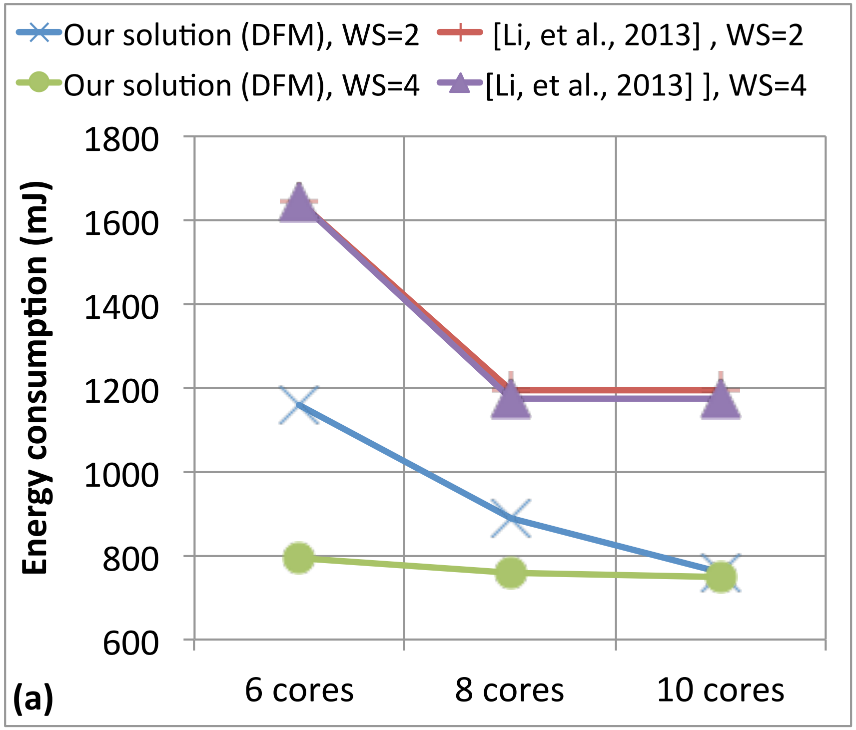
- Adapting *[Wei, et al. 2010]* to the general DAG model by exploiting the output of our DFM



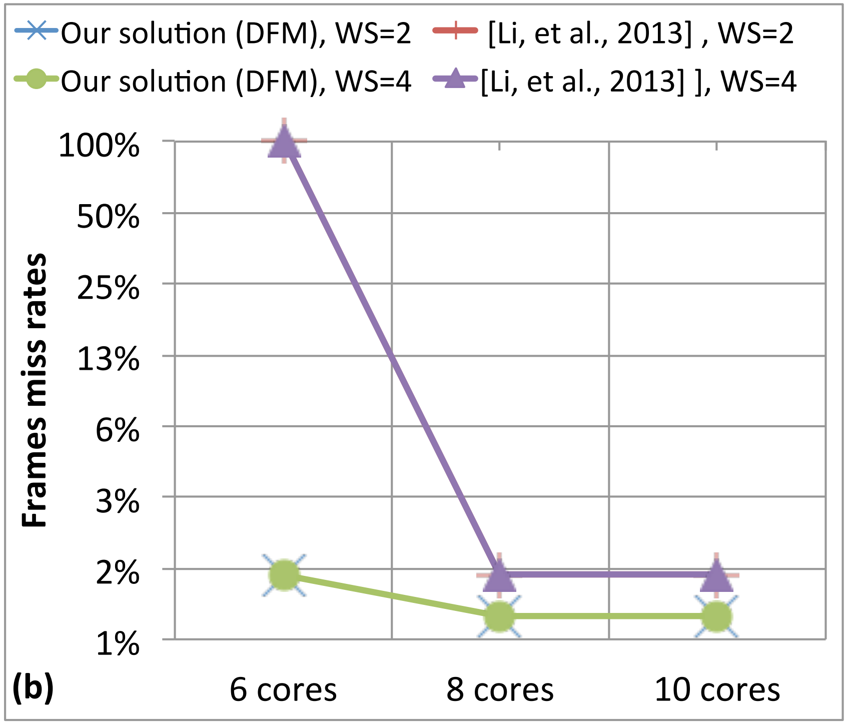
- Bonus: Thanks to our DFM, we can fill the generated gap (filling the gaps with T_{e+l})

H.264 benchmarks
Energy consumption and deadline miss rates

- Variation of the number of deadlines in the buffer and the number of cores
- Up to **52% of energy reduction** and **over 80% reduction in deadline miss rates**



Energy consumption

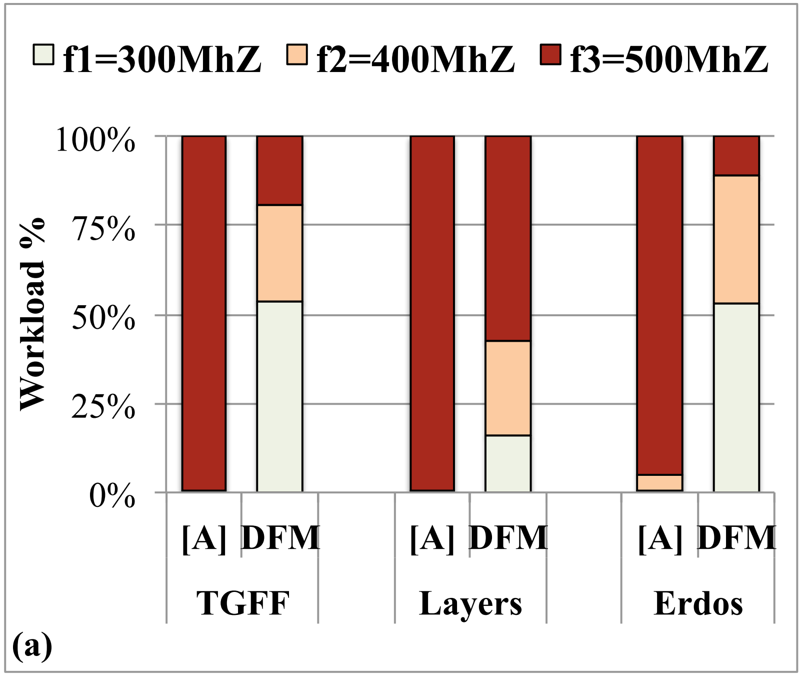


Deadline miss rates

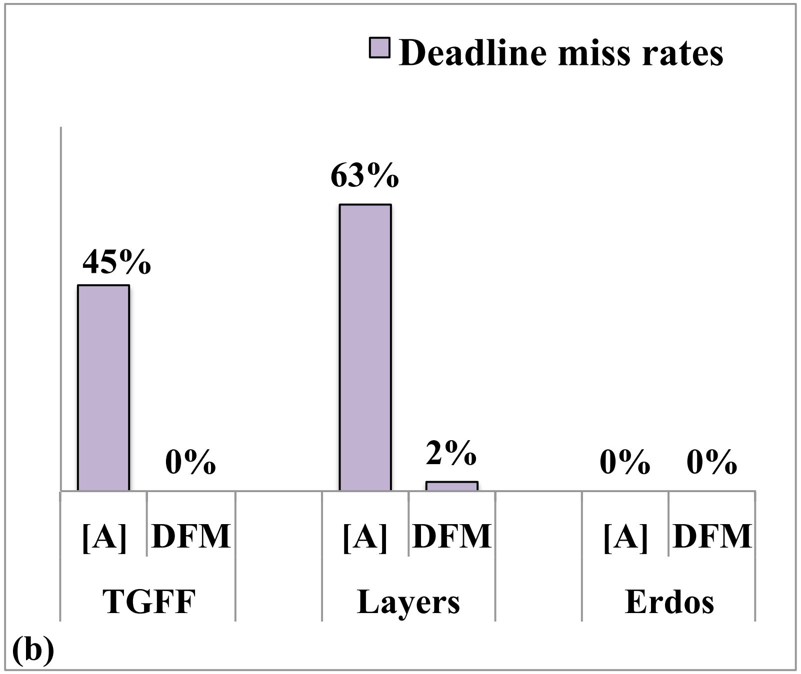
[Li, et al., 2013] *J. Li, et al., "Analysis of Global EDF for Parallel Tasks," in Proc. ECRTS, 2013.*

Synthetic DAG benchmarks
Frequency usage and deadline miss rates

- Deadlines values set 1% greater than the critical path workload (6 cores)
  → **Simulate congested system**
- 25 tasks per deadline; 40% workload variation; buffer size = 4 deadlines
- Up to ***42% of energy reduction*** (using ARM9 power figures)



Frequency usage



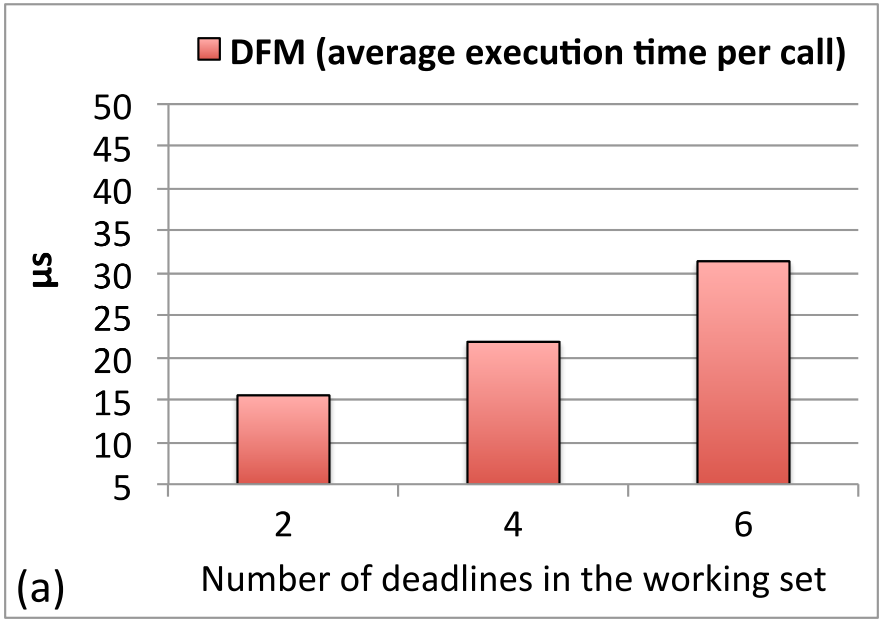
Deadline miss rates

[A] *J. Li, et al., "Analysis of Global EDF for Parallel Tasks," in Proc. ECRTS, 2013.*

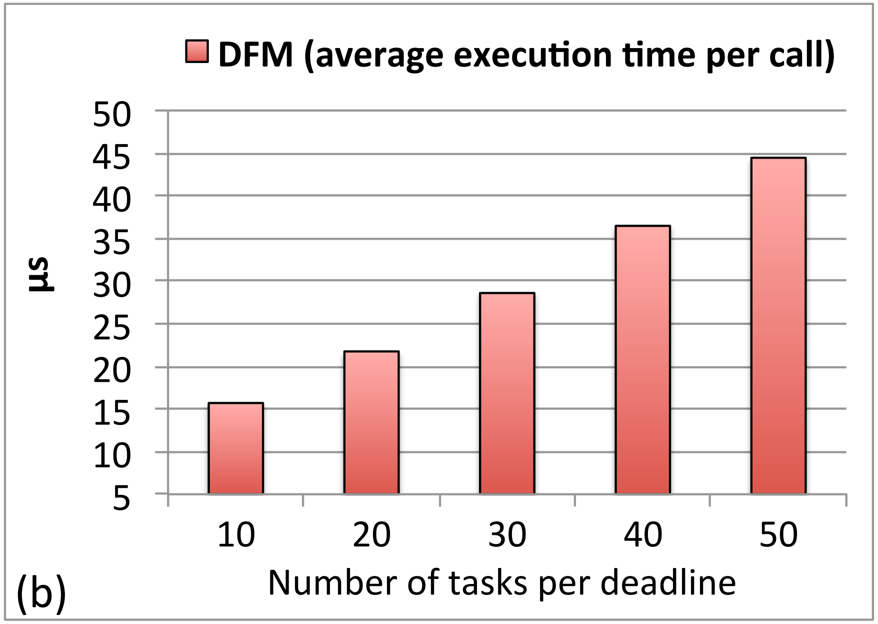
Computation overhead analysis

- Execution time measured on the iPhone 5
- Our H.264 DAG model has 300 tasks per 1 second

→ Our DFM with a buffer size of 4 deadlines needs only ***650μs*** *to proceed the 300 tasks* ***(<<1% overhead)***



H.264 - 20 tasks per deadline



TGFF - 4 deadlines in the buffer

Conclusion

- We have proposed a unified DAG analysis solution
  - Low complexity online solution
  - No restrictions were imposed, covering general DAG models
  - Providing detailed information about the execution status of tasks and deadlines within a look-ahead window
- Significant reduction in energy consumption and deadline miss rates
  - **H.264 video decoder** and **Synthetic DAGs**: up to 52% of energy reduction and over 80% reduction in deadline miss rates
  - **Computation overhead**: *0.65%* overhead (H.264 application)

Thank You

QUESTIONS ?

Karim.Kanoun@epfl.ch
Embedded Systems Laboratory
EPFL, Switzerland