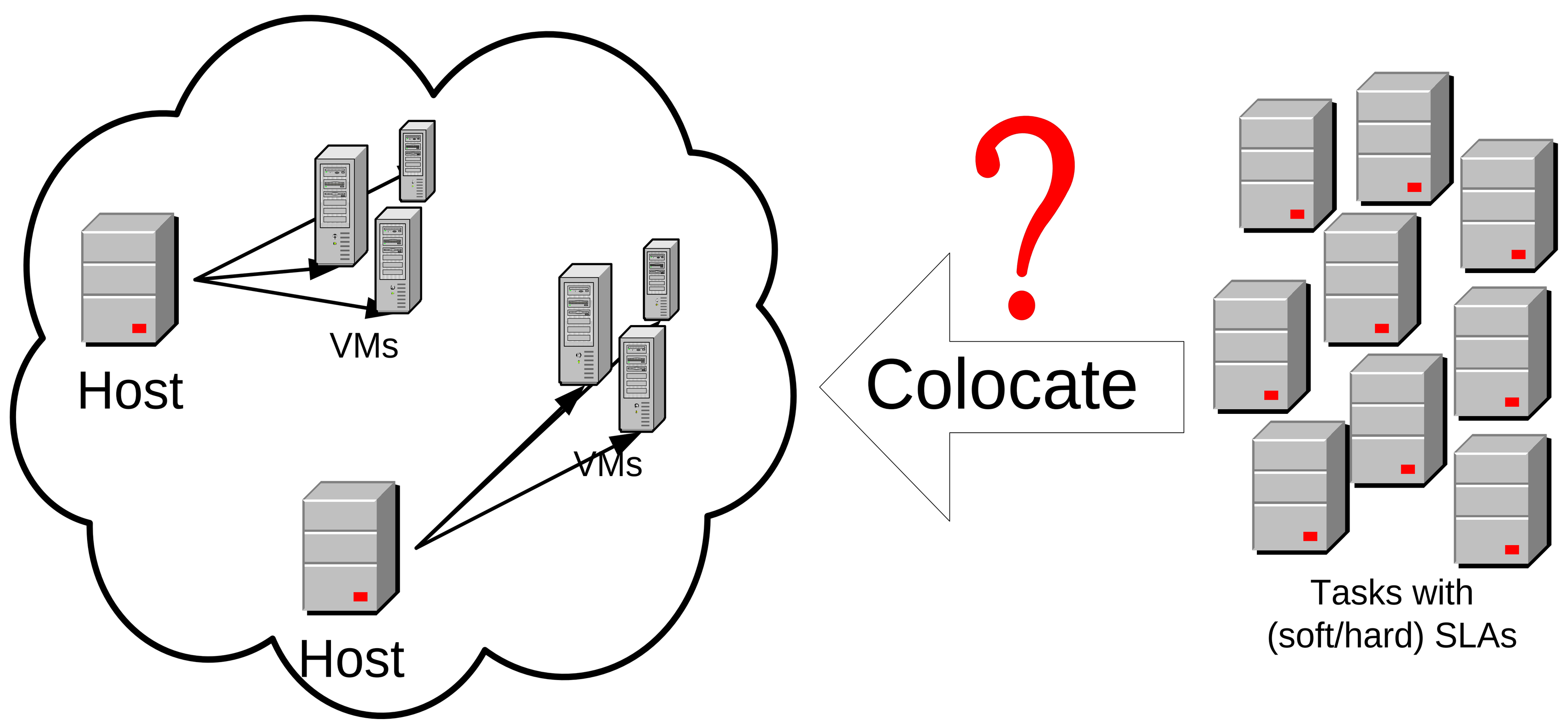
A Type-Theoretic Framework for Efficient and Safe Colocation of Periodic Real-time Systems

Vatche Ishakian, Azer Bestavros, and Assaf Kfoury

Motivation



Previous Work

- Multiprocessor Scheduling
  - NP-Hard and heuristics inefficient
- Hierarchical Scheduling
  - Assumes knowledge of mapping

Not schedulable set of tasks

C	1	2	3	4	5
T	4	9	17	34	67

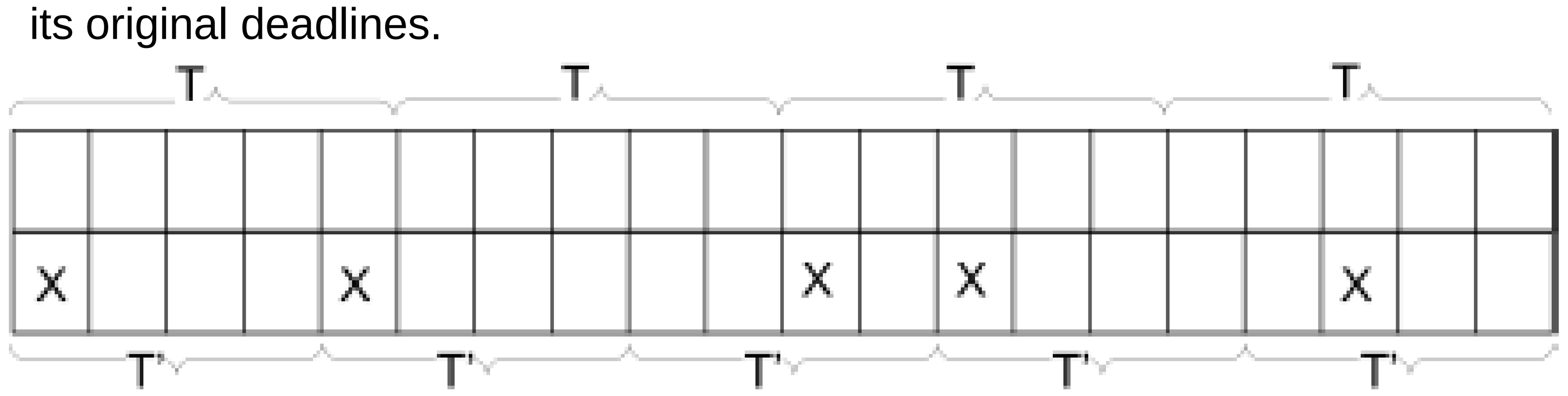
Schedulable set of tasks

C	1	2	3	4	5
T	4	8	16	32	64

- Using safe Transformation (e.g.. more frequent allowance)

Unsafe Transforms

Consider a task that requires C = 1 time units of the resource every period T = 5 time units. While reducing the allocation period for this task from T = 5 to T′ = 4 would result in that task being allotted the resource for a larger fraction of time (25% as opposed to 20%), it is possible for that task to miss its original deadlines.



The upper row shows the periodic boundaries as originally specified (T = 5), whereas the lower row shows a periodic allocation with (T′ = 4), with "X" marking the times when the resource is allocated

Why Safe Transforms?

- Leverage flexibility of Soft SLA's
- Benefit from frequent allowance
- **Contribution**: Defined a and proved a set of safe transformations.

May need Multiple Transforms

- Large solution search space
- Transformation composition not necessarily transitive.
- **Contribution**:
  - Defined and proved a safe transform composition.
  - Efficient search heuristic

A Type-Theoretic Transform Framework

- Defined as a quadruple (C,T,D,W) where
  - C is the capacity of the resource
  - T is An allocation interval
  - D is the maximum number of allowed misses
  - W is a window consisting of W > 1 allocation intervals.
- Flexible enough to model both soft and hard real time requirements.
- Allows for new transforms to be codified
- Allows for mapping inference

Uniprocessor Results

Hard SLAs

Overload	Success Rate
0 - 7%	52.27%
7% - 14%	26.27%
14% - 21%	7.73%
21% - 28%	1.47%
28% - 35%	0.13%

Soft SLAs

Overload	Success Rate
0 - 7%	90.00%
7% - 14%	66.53%
14% - 21%	48.93%
21% - 28%	28.93%
28% - 35%	17.07%

- Results highlight the success rate of our heuristic given the overload

Multiprocessor Results

- We use the schedulability condition from Andersson et al [1]. Any number of arbitrary tasks can be scheduled on m identical multiprocessors if sum of tasks utilization $U(t) < m^2/3m - 2$
- If the tasks were harmonic, then the bound would be $U(t) < m^2/2m - 1$

Proc Before	Success Rate	Avg Tasks	Proc After
3	94.89%	4.67	2
4	96.67%	5.84	2.03
5	99.33%	6.92	2.72
6	99.78%	8.25	3.04
7	100.00%	9.64	3.48
8	100.00%	10.88	4.08
9	100.00%	12.1	4.44
10	100.00%	13.63	5.05
11	100.00%	15.1	5.48
12	100.00%	16.28	6.1
13	100.00%	17.71	6.6
14	100.00%	18.82	7.12
15	100.00%	20.36	7.65
16	100.00%	21.4	8.24

- As the overload increases, the chances of finding a feasible transformation decreases.
- With the use of transformations, we are able to decrease the number of processors needed to support the workload's SLA by a factor of two.

References

[1] B. Andersson, S. Baruah, and J. Jonsson, Static-priority scheduling on multiprocessors, in RTSS 2001

[2] Vatche Ishakian, Azer Bestavros, and Assaf Kfoury. A Type-Theoretic Framework for Efficient and Safe Colocation of Periodic Real-time Systems, Tech. Report 2010-002, BU, CS Dept, 2010.