

Analytically Bounding Data Cache Behavior for Real-Time Systems



Harini Ramaprasad and Frank Mueller



1 Motivation

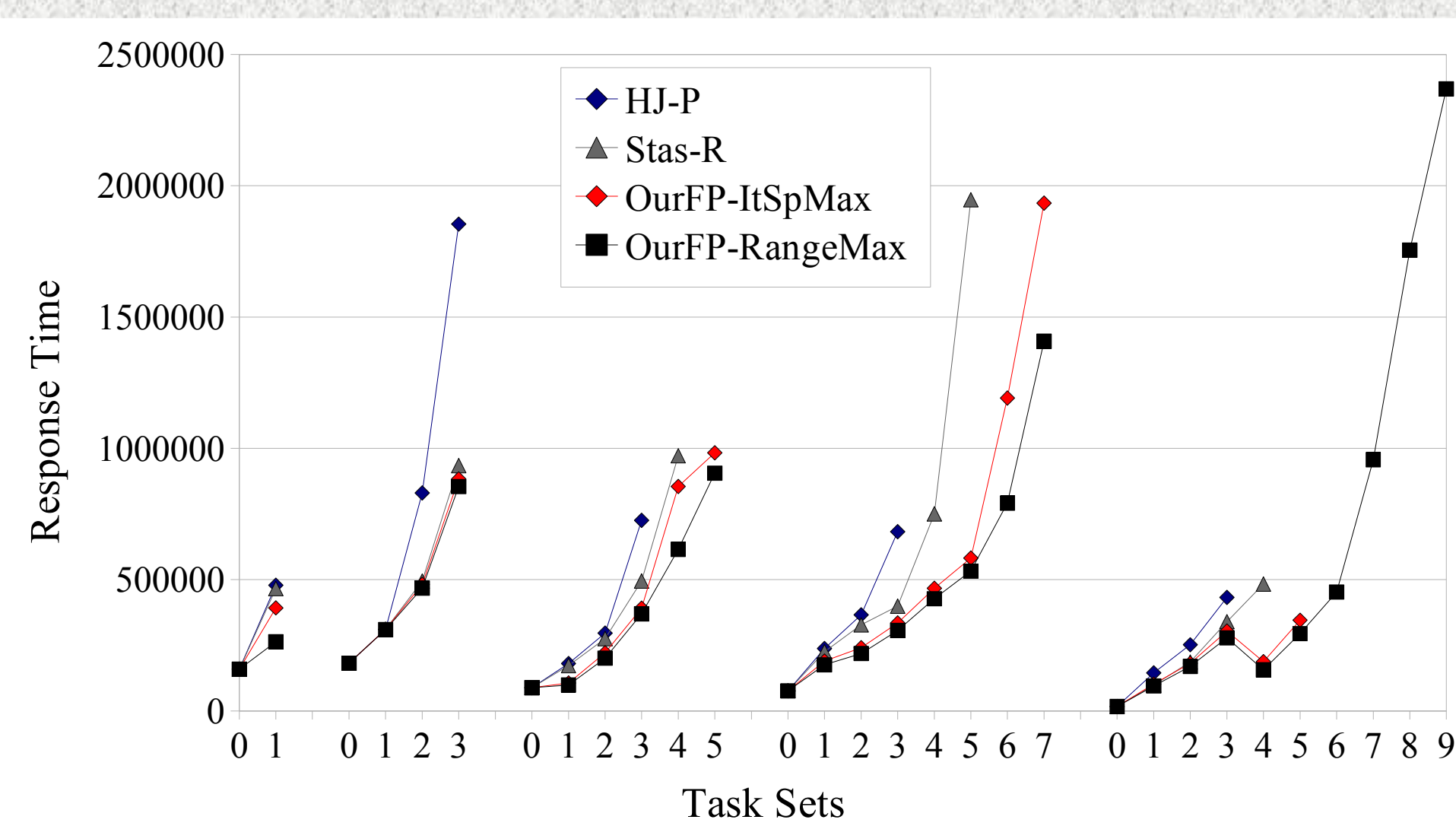
- Real-time systems have **temporal** constraints
 - A-priori* schedulability guarantee required
- Schedulability analysis of real-time systems
 - Worst-case execution time (WCET) of tasks required
 - Static timing analysis (STA) used for this
- Data Caches (D\$)
 - Difficult to characterize** → *unpredictability* in STA
 - Add to pessimism** of WCET estimates
- Getting rid of data caches causes performance loss!
- Solution: Try to analyze and predict data cache behavior

2 Single Task Example

- Input
 - $A[1..5][1..5]$ – base address = 151944
 - $B[1..5][1..5]$ – base address = 153068
 - Loop nest:

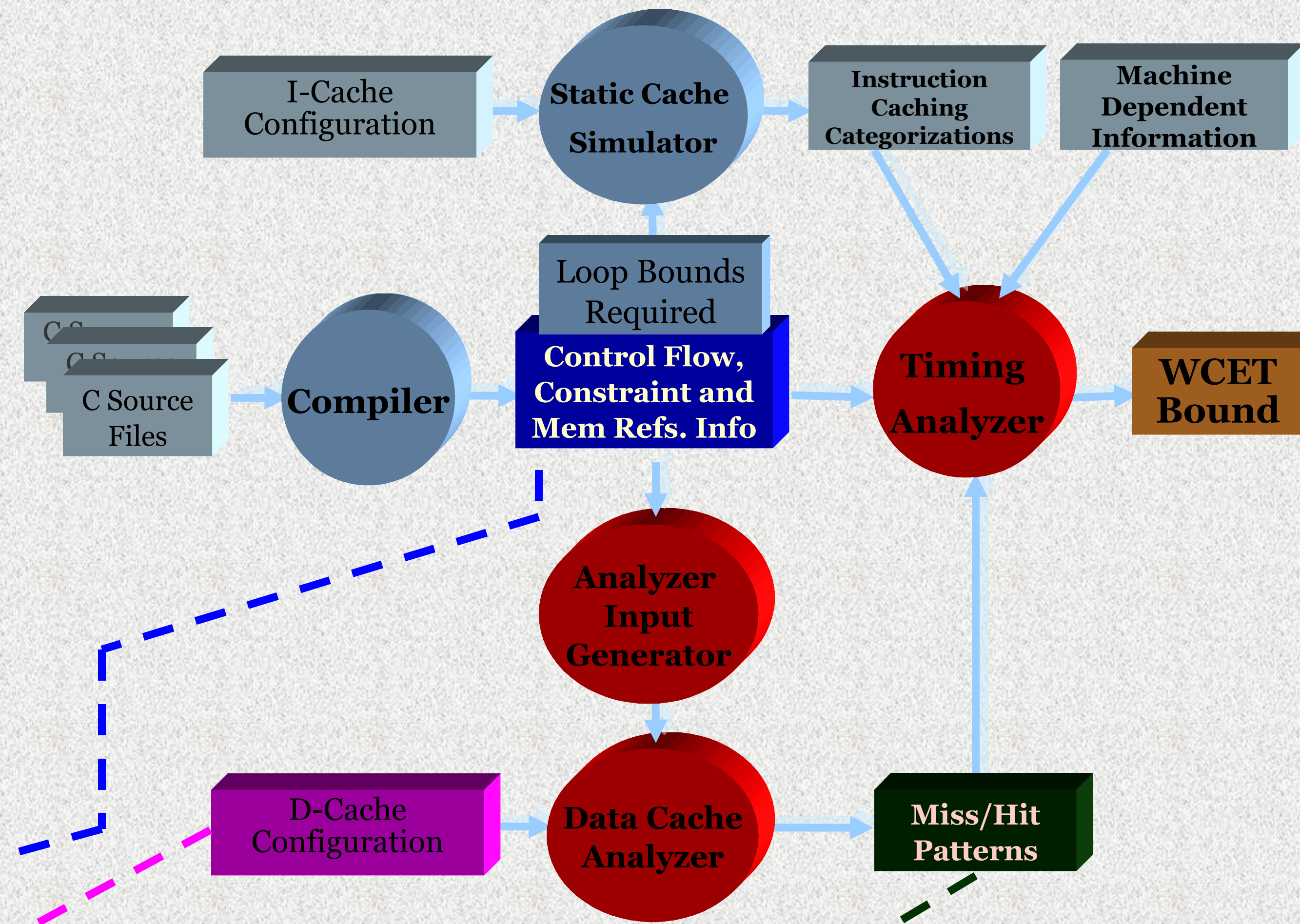

```
for(r = 0; r < 5; r++)
  for(c = 0; c < 5; c++)
    A[r][c] = A[r][c] + B[r][c]
```
- Cache configuration
 - 32 byte cache line, 1 KB cache, direct mapped
- Our output
 - Ref 1: MhMMhhhhhhhhhhMhhhhMhhhhM – total misses = 6
 - Ref 2: MMhMhhhhhhMhhhhMhhhhMhhhh – total misses = 6
 - Ref 3: hhhhhhhhhhhhhhhhhhhhhhhhhhh – total misses = 0

3 Results (U = 80%)



- Response time increases with decrease in priority
- Our method has **least rate of increase**
- All task sets deemed **schedulable** by our method

Static Timing Analysis Framework



4 Single Task Analysis

- Analyze loop-nest oriented code
- Enhance **Cache Miss Equations (CME)** framework
- Allow arbitrary loop nests
 - Forced loop fusion**
 - Concatenate** iteration spaces of sequential loops
 - Permit non-rectangular loops
- Derive exact D\$ reference patterns
 - Miss/Hit patterns for each reference**

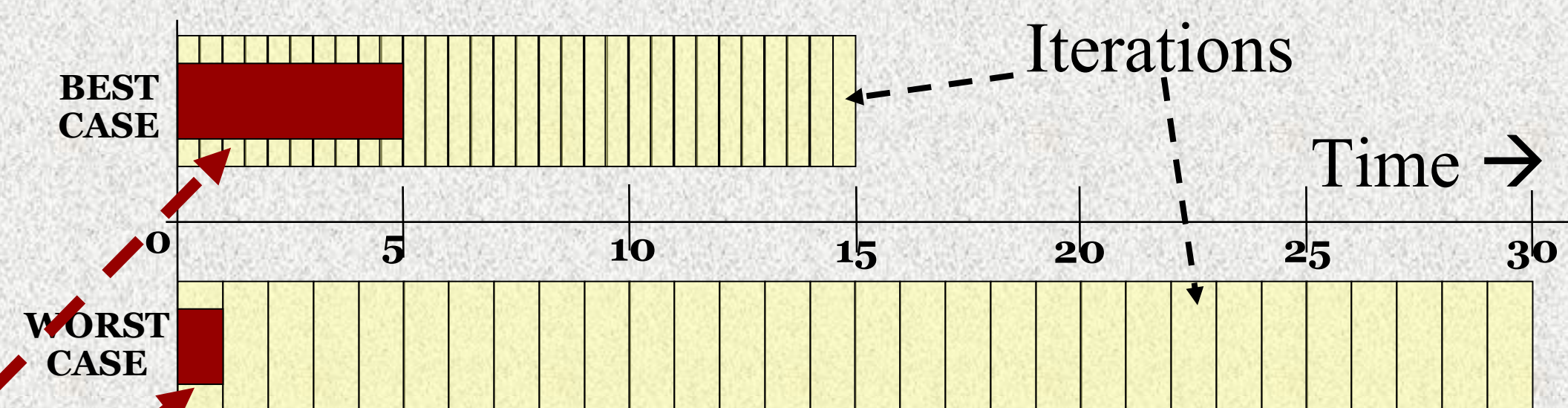
5 Multiple Task Example

Task	Period	WCET	BCET
T ₀	20	7	5
T ₁	50	12	10
T ₂	200	30	15

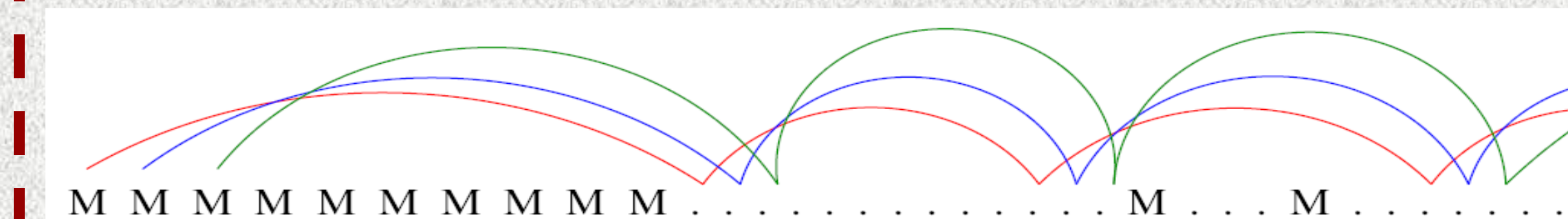
6 Multiple Task Analysis

- Multiple *prioritized* tasks share D\$ → preemption occur
- Data of preempted task may be evicted from D\$
- D\$ related preemption delay (**D-CRPD**) required
- Response time** of task requires *upper bound* on
 - Number of **feasible** preemption points
 - Use execution range (BCET-WCET) of tasks
- Progress in execution of task at each point
 - Map **execution time** range of task to **iteration** range
- Preemption delay at each point
 - Use **access chains** (represent memory accesses)

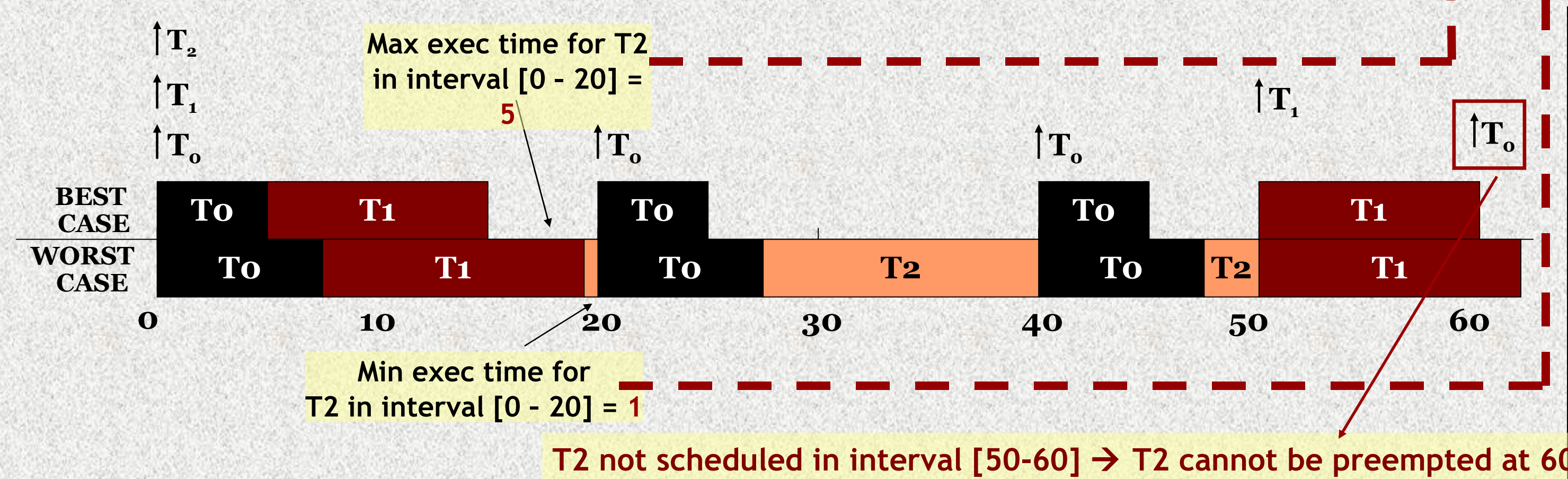
B. Map time range to iteration range



C. Calculate delay using Access Chains



A. Feasible preemptions for task T₂



7 Conclusions

- Schedulability analysis for hard real-time systems**
- Single task analysis
 - Hit/miss pattern* → number & position of misses
 - Safe and tight** WCET estimate
- Multiple task analysis
 - Eliminate infeasible preemptions
 - Tight** WCET & response time estimates
 - Increased schedulability of task sets