

Uneven Memory Regulation for Scheduling IMA Applications on Multi-core Platforms

Problem description

- Schedule IMA applications on multicore platform with shared channels
- Ensure temporal behavior in spite of interference from sharing channels
- Any misbehaving application should not affect the execution of other applications

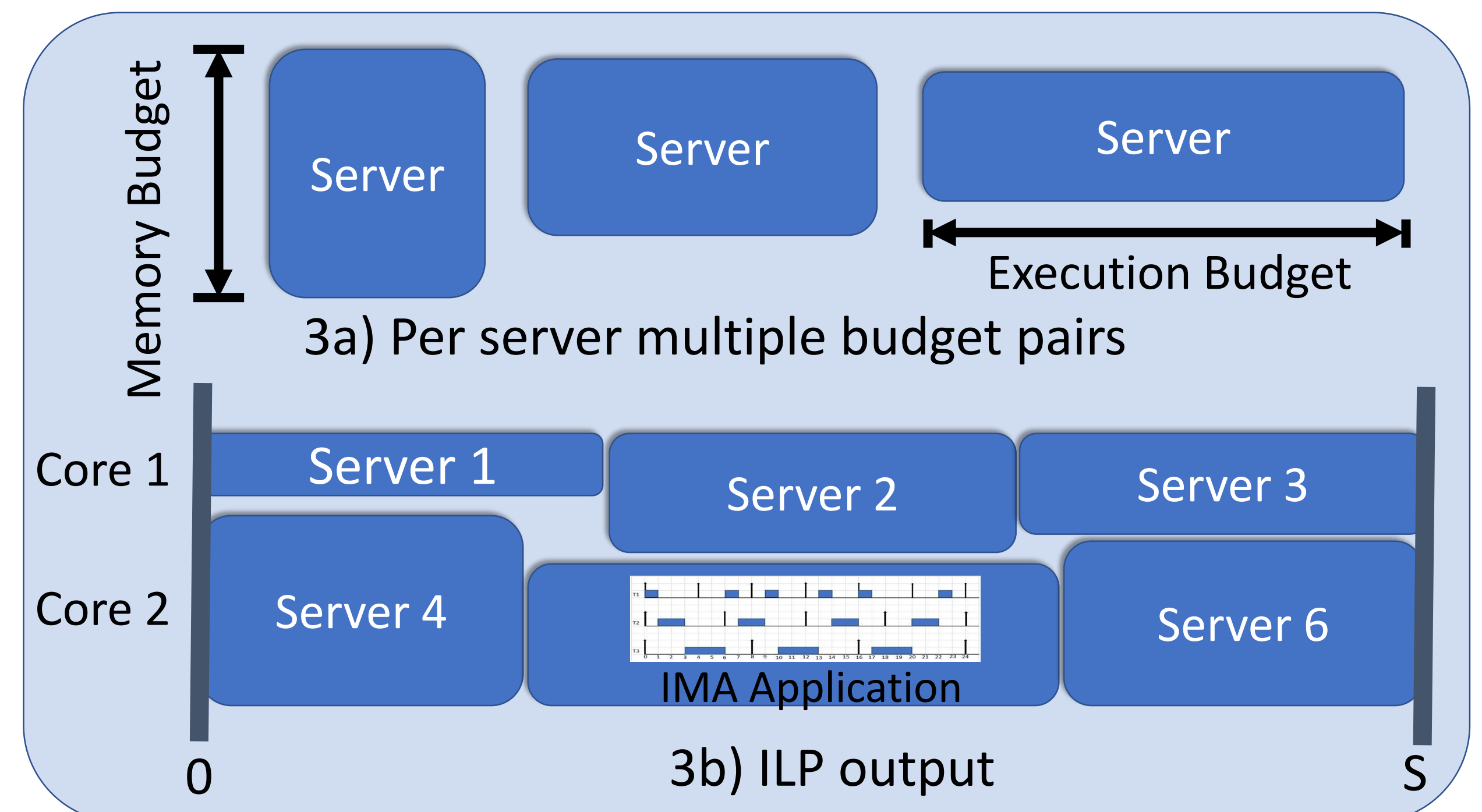
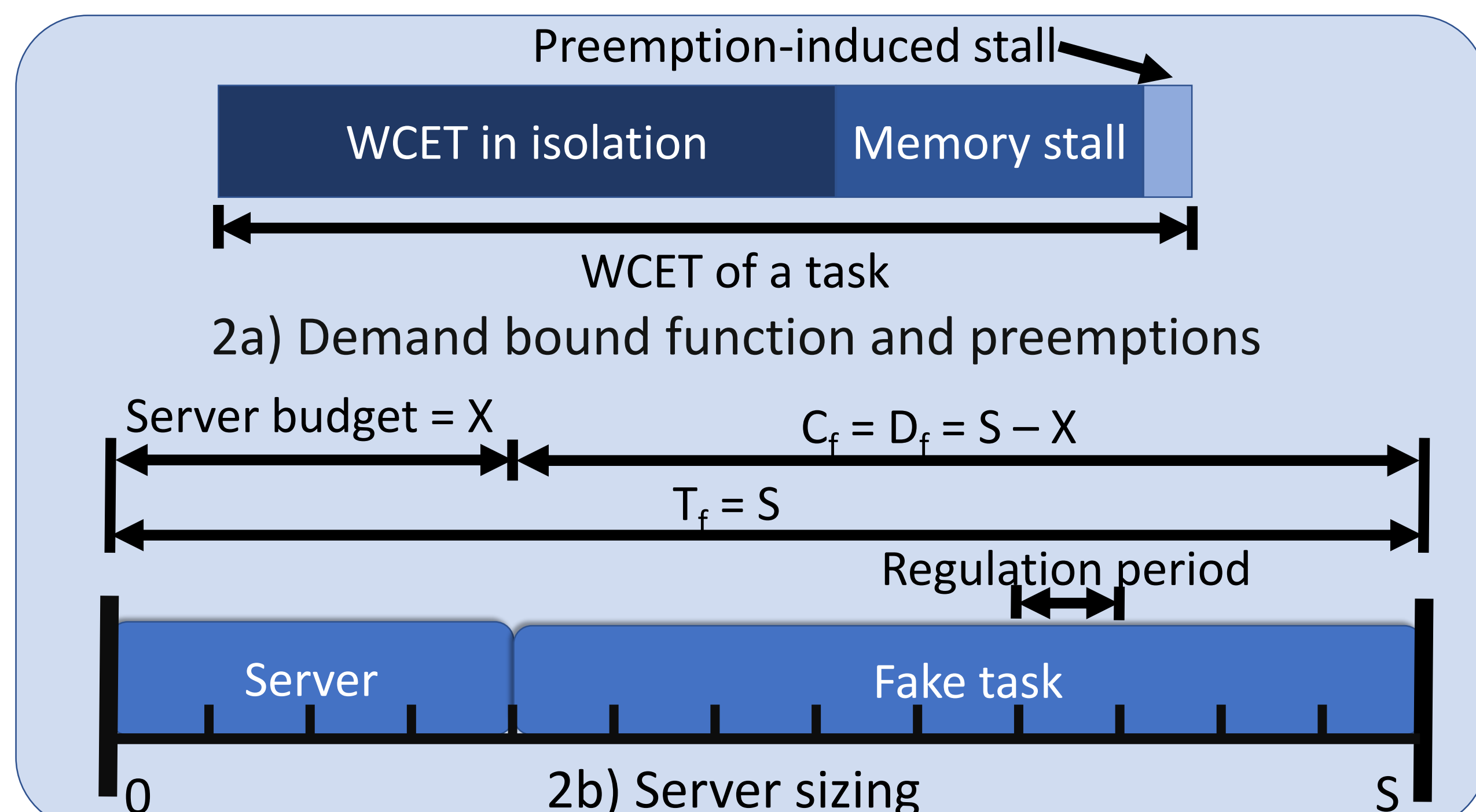
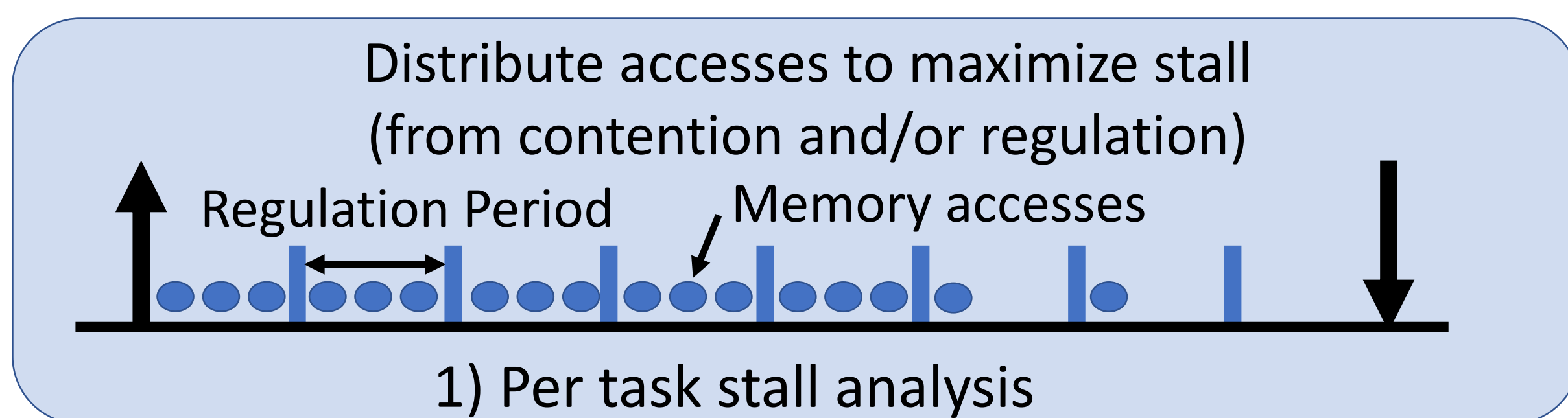
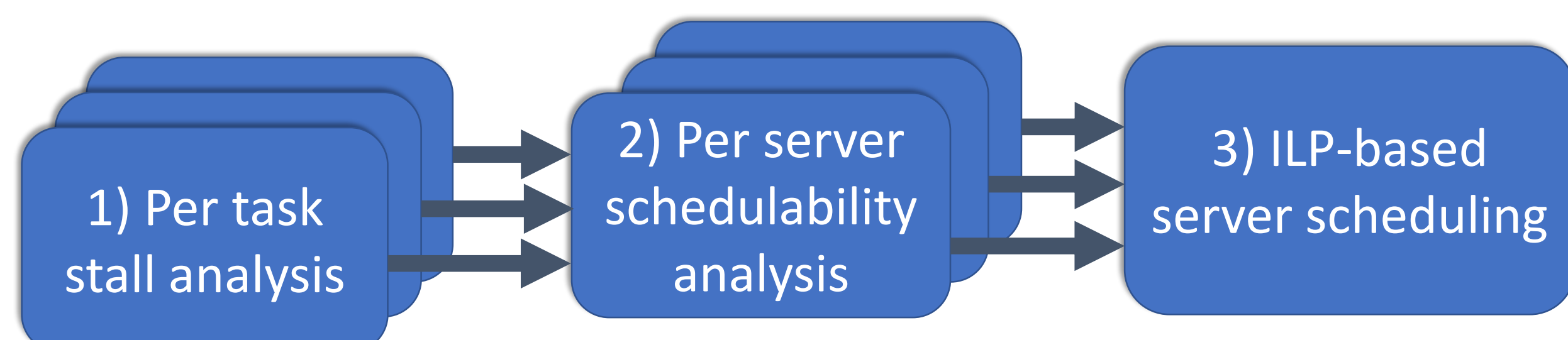
Main idea

- Mitigation of interference on shared channels (memory) on multicore platform via memory regulation
- Integration of stall due to inference on shared resources into the schedulability analysis
- Isolation among applications on each core via servers
- Assign execution and memory budgets to each server

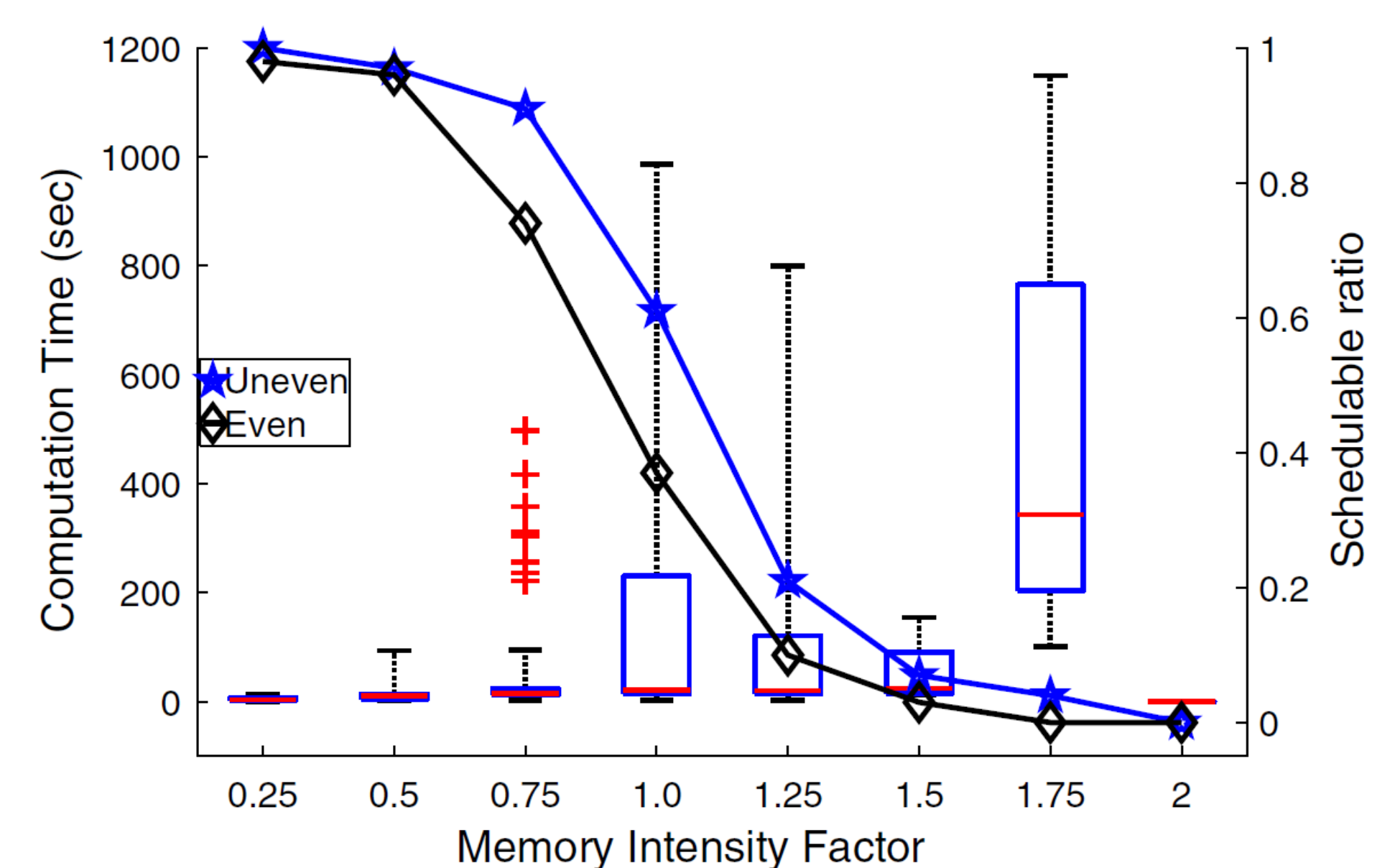
System model

- Inline with Single-Core-Equivalence framework assumptions
- Isolates applications via servers
- Cyclic executive schedule for servers
- Earliest Deadline First scheduling inside each server
- Memory bandwidth regulation via MemGuard
- Uneven memory-regulation budget per server
- Identical multicore platform
- Round robin arbitration at memory bus and controller level
- Multiple outstanding memory requests

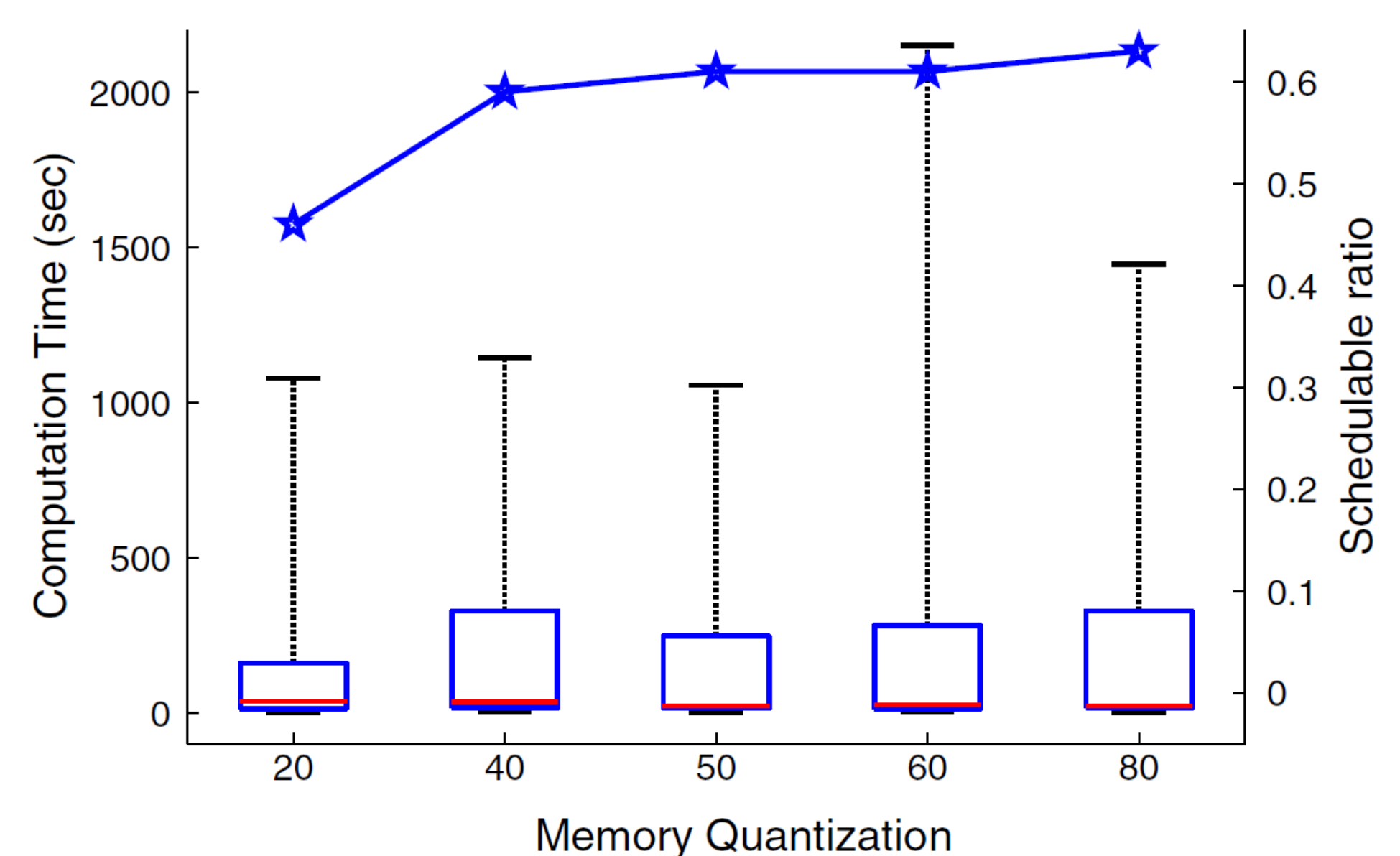
Our approach



Evaluation



Uneven vs. even regulation schedulability ratio



Analysis computation time for a 4-core platform

Concluding remarks

- Allows uneven memory bandwidth per server
- New stall and schedulability analysis for server-based approach
- ILP model to find feasible cyclic executive schedule of servers
- Uneven per-server memory bandwidth significantly improve schedulability with realistic resource demand
- Tractable solution for multicore platforms up to four cores