Dynamic Reconfiguration for Adaptive Computing in Heterogeneous SoCs

Prabhat Mishra
University of Florida

Umit Ogras
Arizona State University

Introduction

The goal of this project is to develop an efficient dynamic reconfiguration framework to enable adaptive computation, communication and storage in heterogeneous multicore SoCs under energy, performance, reliability, thermal and real-time constraints.

- Reconfiguration Primitives
  - Dynamic voltage and frequency scaling
  - Dynamic task mapping to CPU/GPU cores
  - Dynamic NoC reconfiguration
  - Dynamic cache reconfiguration
- Objectives / Constraints
  - Power, performance, energy, reliability, temperature and real-time constraints

Motivation

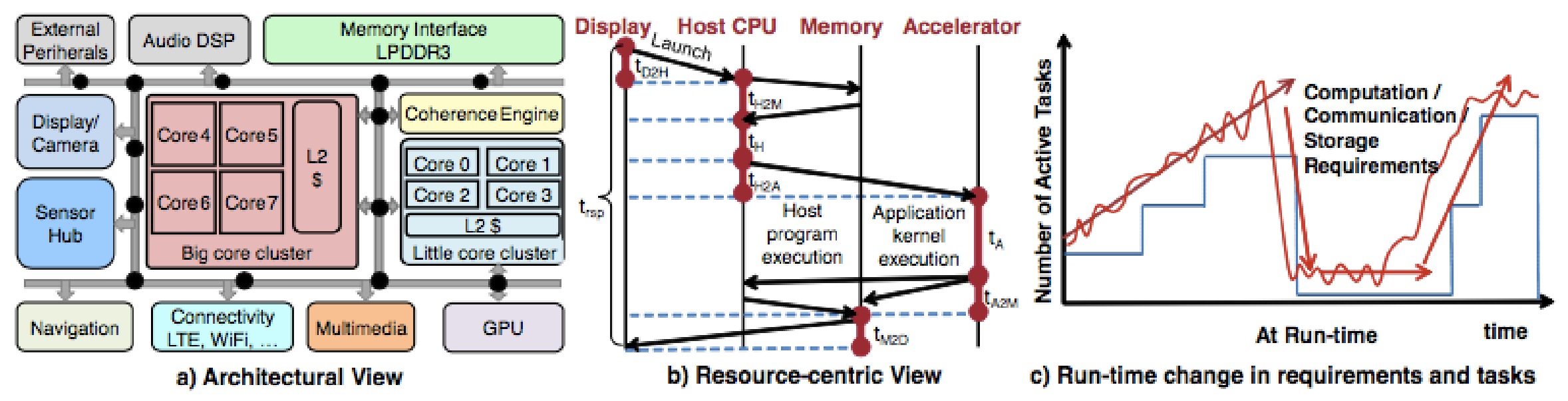


Fig.1: (a) Heterogeneous MpSoCs, (b) application flow, and (c) run-time change in requirements.

- Embedded systems are highly heterogeneous computing platforms, which include multiprocessor system-on-chip (MpSoC), touch displays, modems, flash memory, camera, GPS modules, etc.
- MpSoC itself consists of many different processing elements (PE) including multiple CPU cores, graphical processing units (GPU), DSP cores and video accelerators, as shown in Fig.1(a).
- Only a subset of the resources are invoked during the lifetime of an application. For example, a navigation application goes through a number of phases as illustrated in Fig. 1(b).

System Modeling

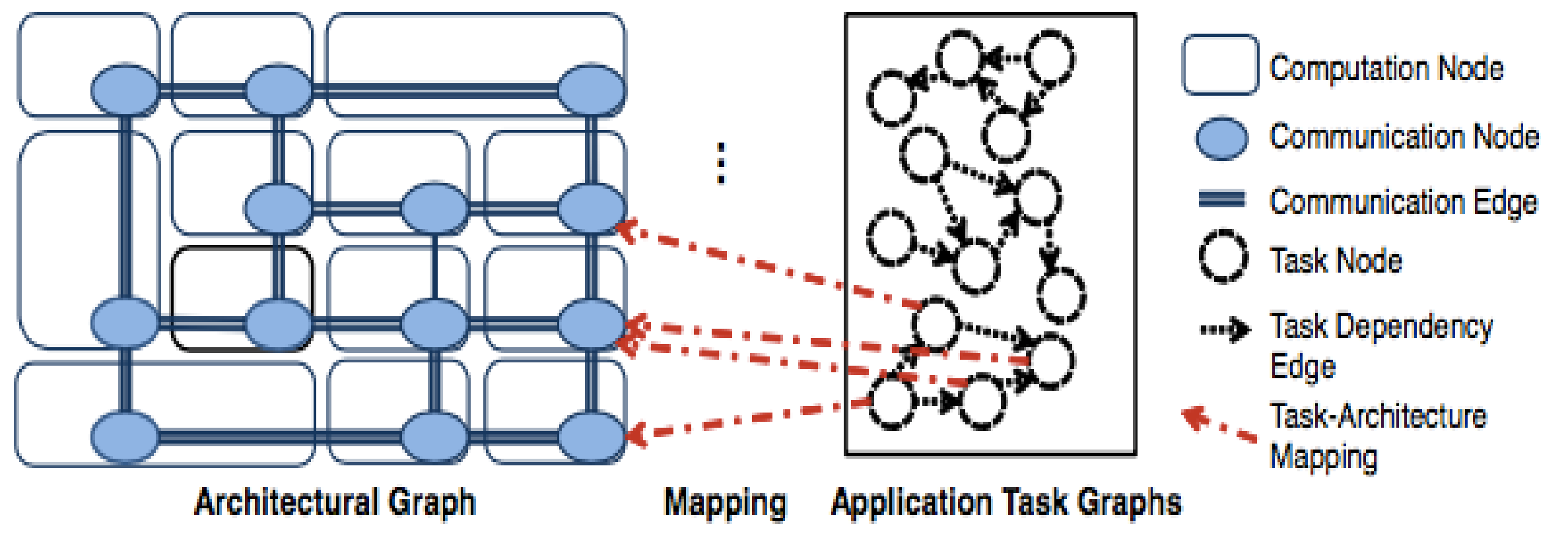


Fig.2: Modeling of architecture, tasks and their mappings.

- The task model is a graph where the nodes are the set of tasks and edges indicate communication between tasks.
- The tasks can be dependent or independent, periodic or aperiodic with soft or hard deadlines.

Dynamic Cache Reconfiguration

DCR in Multicore SoCs:

- Multicore processors impose unique opportunities as well as challenges in applying cache reconfiguration.
- We consider a general cache hierarchy in which each core has its own L1 private cache and groups of cores can share caches (e.g., L2) at different levels.
- L1 caches DCRhave impact on L2 CP in performance / energy / vulnerability as in Fig. 4.

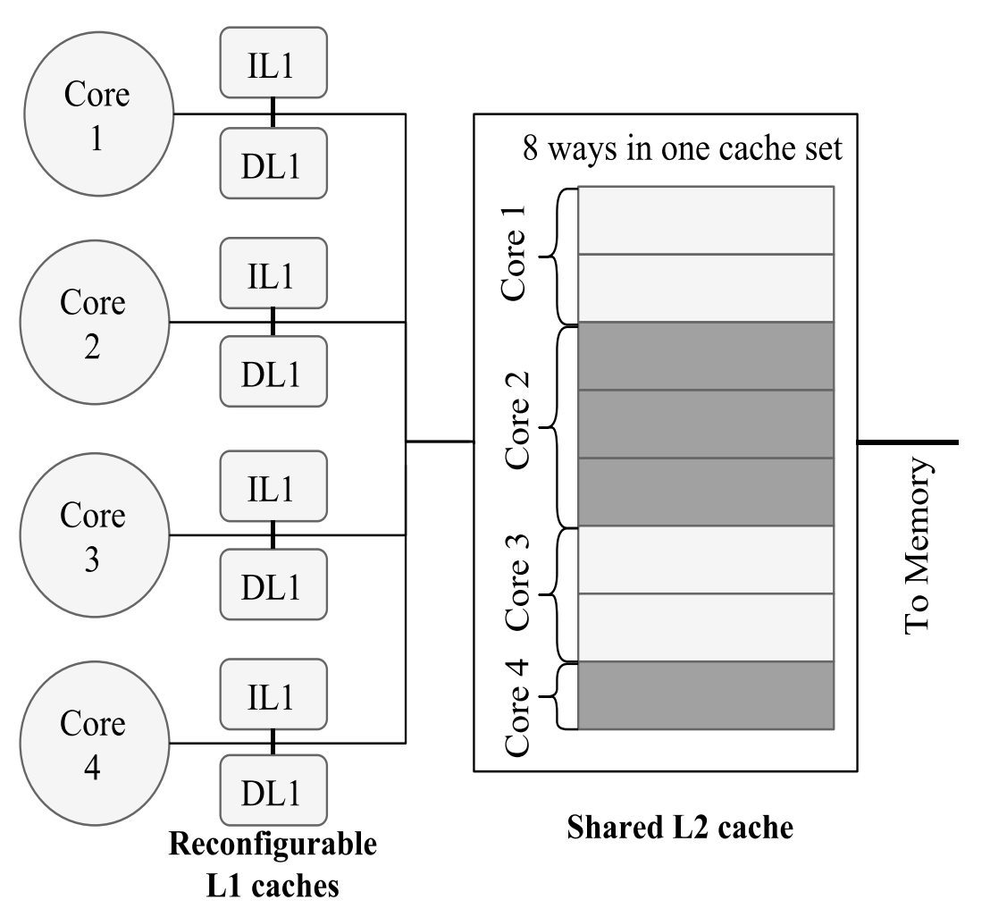


Fig. 3: A multicore system with Reconfigurable L1 caches and way-partitioned L2 cache.

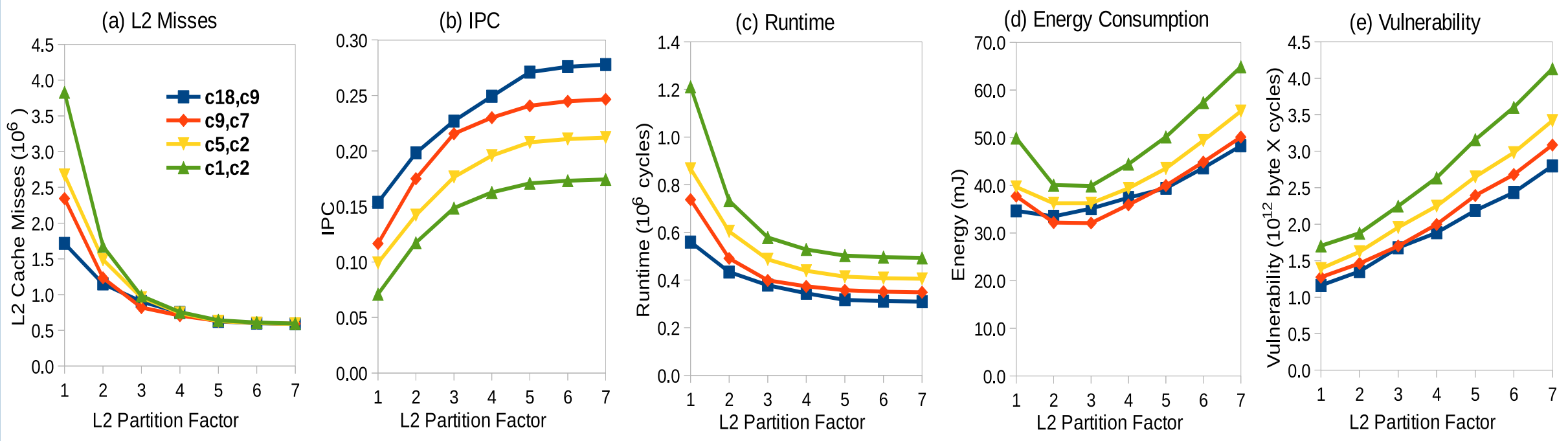


Fig. 4: Inter-dependence of L1 DCR and L2 CP on (a) L2 Misses, (b) IPC, (c) Runtime, (d) Energy and (e) Vulnerability.

We proposed a reconfiguration approach for multicore SOCs:

- DCR on L1 private caches
- CP (cache partition) on the shared L2 cache
- the approach (Fig. 5) works in three steps: (1) static task profiling on each core (2) dynamic algorithm to optimize on each core (3) optimize among all cores to find the partition scheme among all cores.

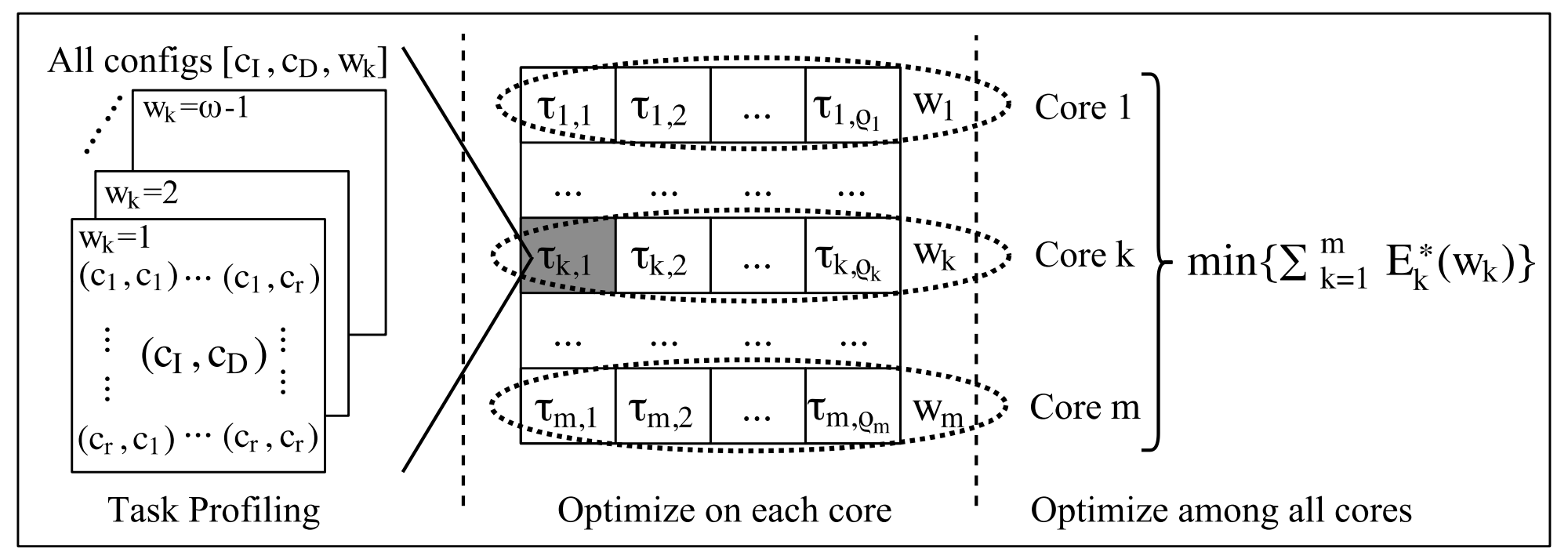


Fig.5: Three-step optimization for L1 DCR + L2 CP

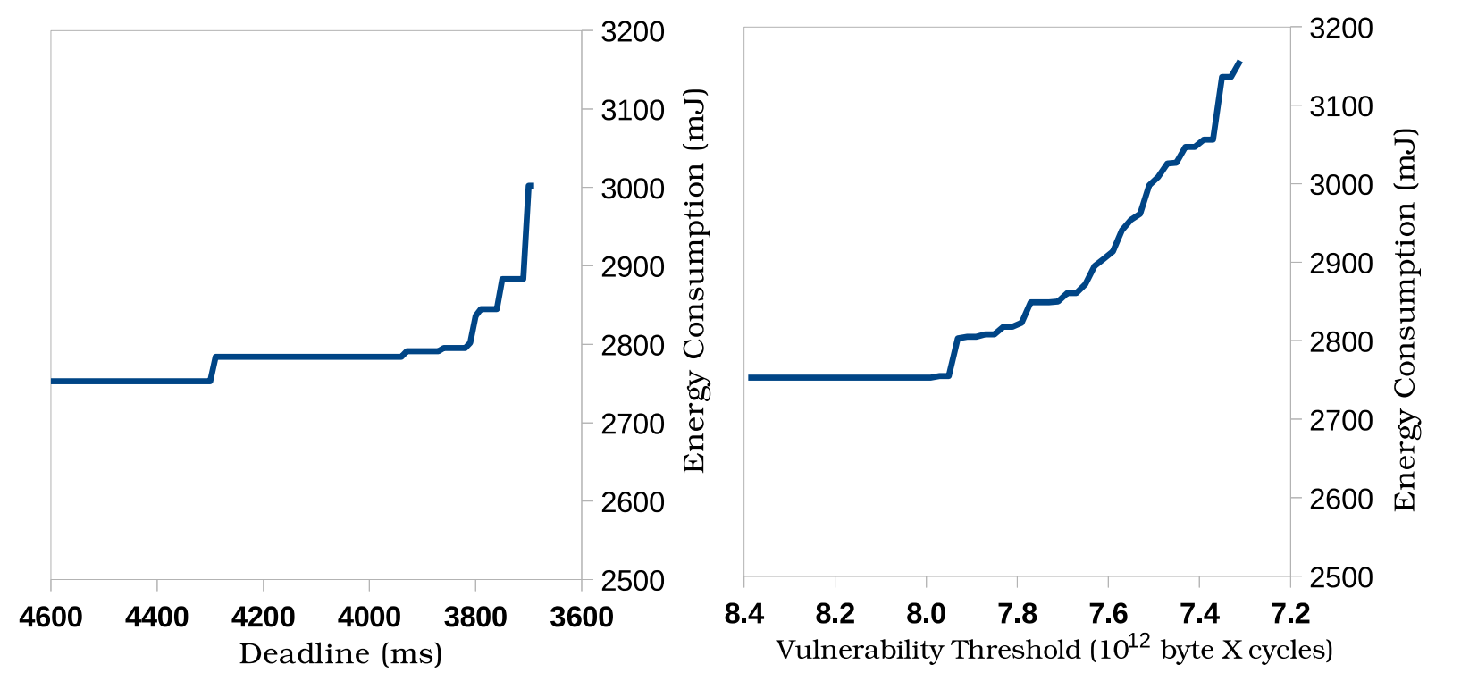


Fig.6: Effects of deadline and vulnerability threshold on Core 1 for Task Set 1. Energy consumption is smaller when threshold is not so strict. If threshold is too strict for deadline or vulnerability, the optimization for this core can not find a solution.

Experimental Results:

- Deadline and vulnerability threshold has impact on the optimization process. (Fig. 6)
- Compared with the state-of-art, we gain significant vulnerability reduction (on average 49.3%) with minor energy overhead (on average 5.6%). (Fig. 7)

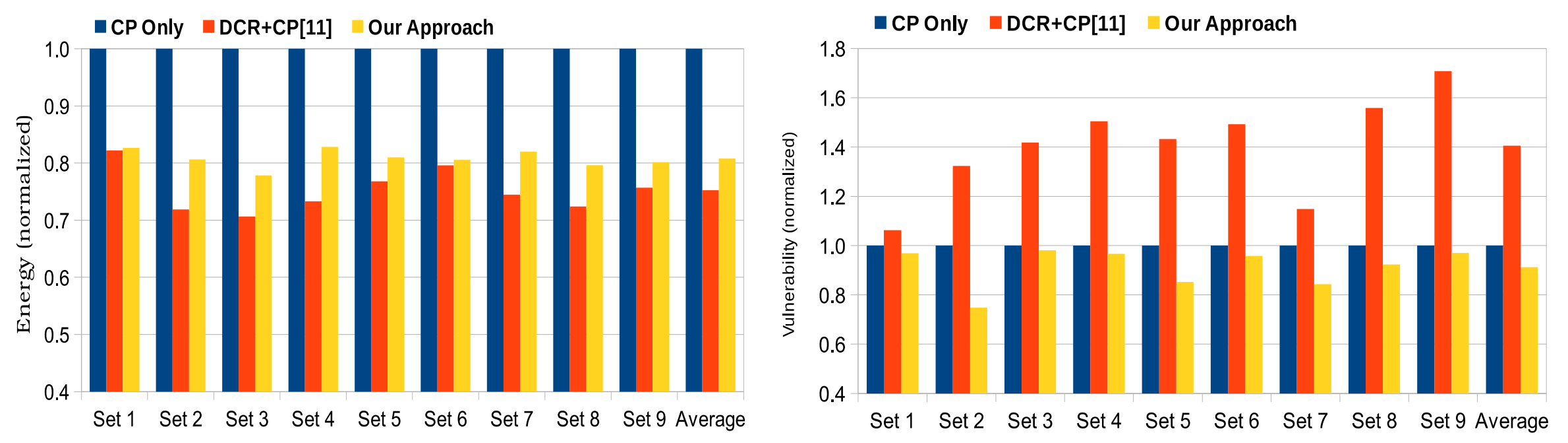


Fig. 7: Comparison of energy consumption and vulnerability of different task sets.

Dynamic Power Budgeting

- MpSoCs contain several processing elements (PE)
- Graphics applications demand high performance
- High performance can lead to thermal violations, as in Fig. 8.
- Power budget needs to be dynamically allocated to each PE

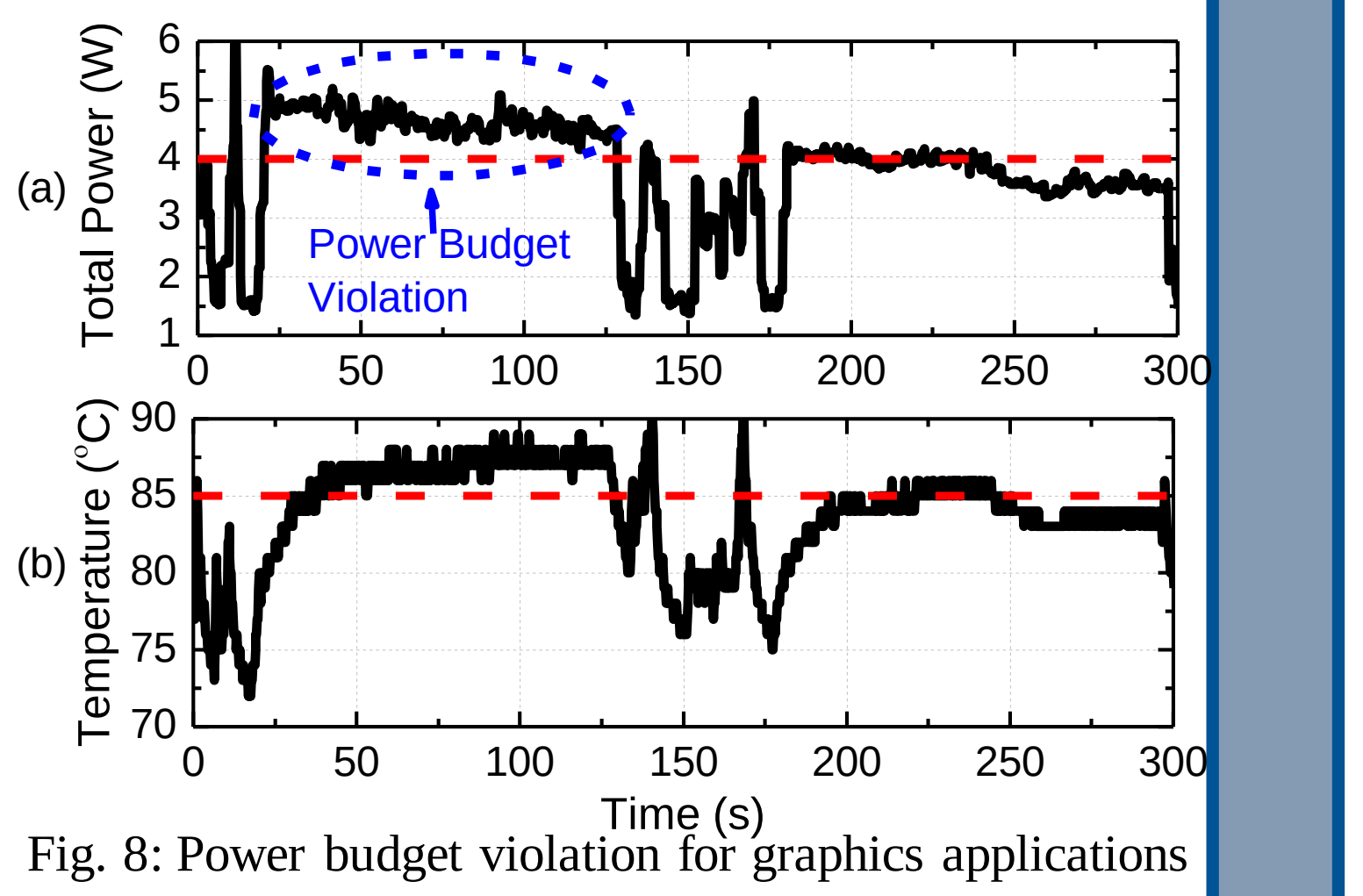


Fig. 8: Power budget violation for graphics applications

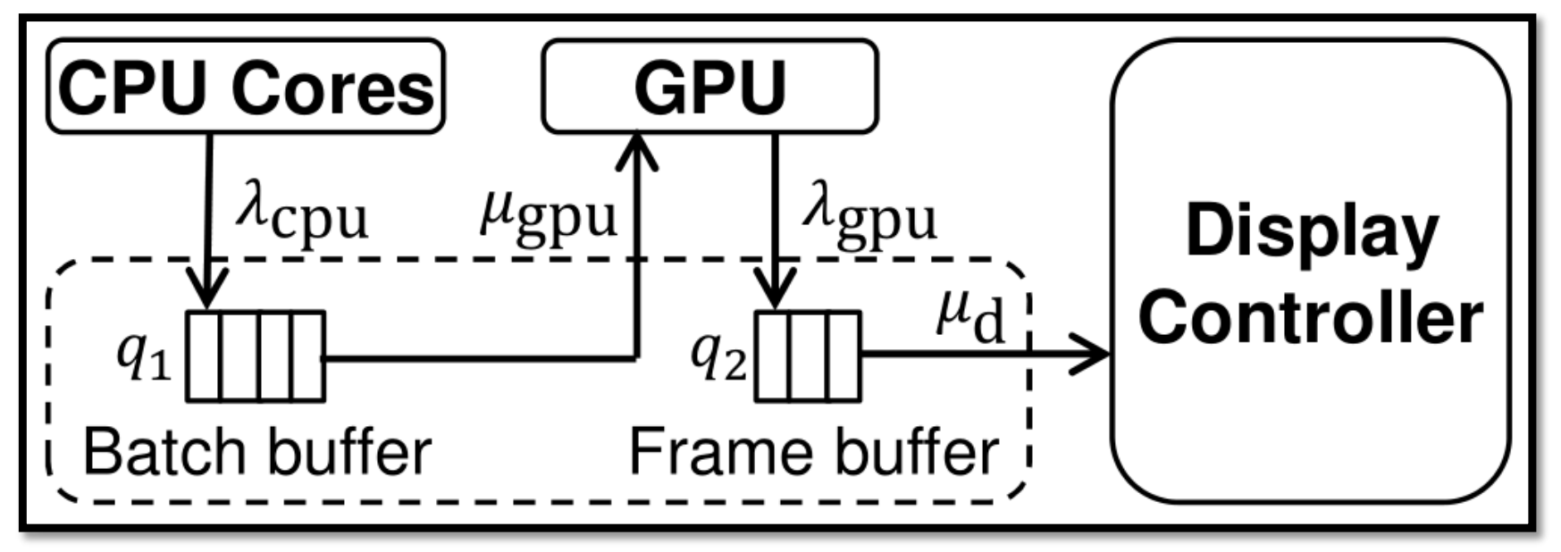
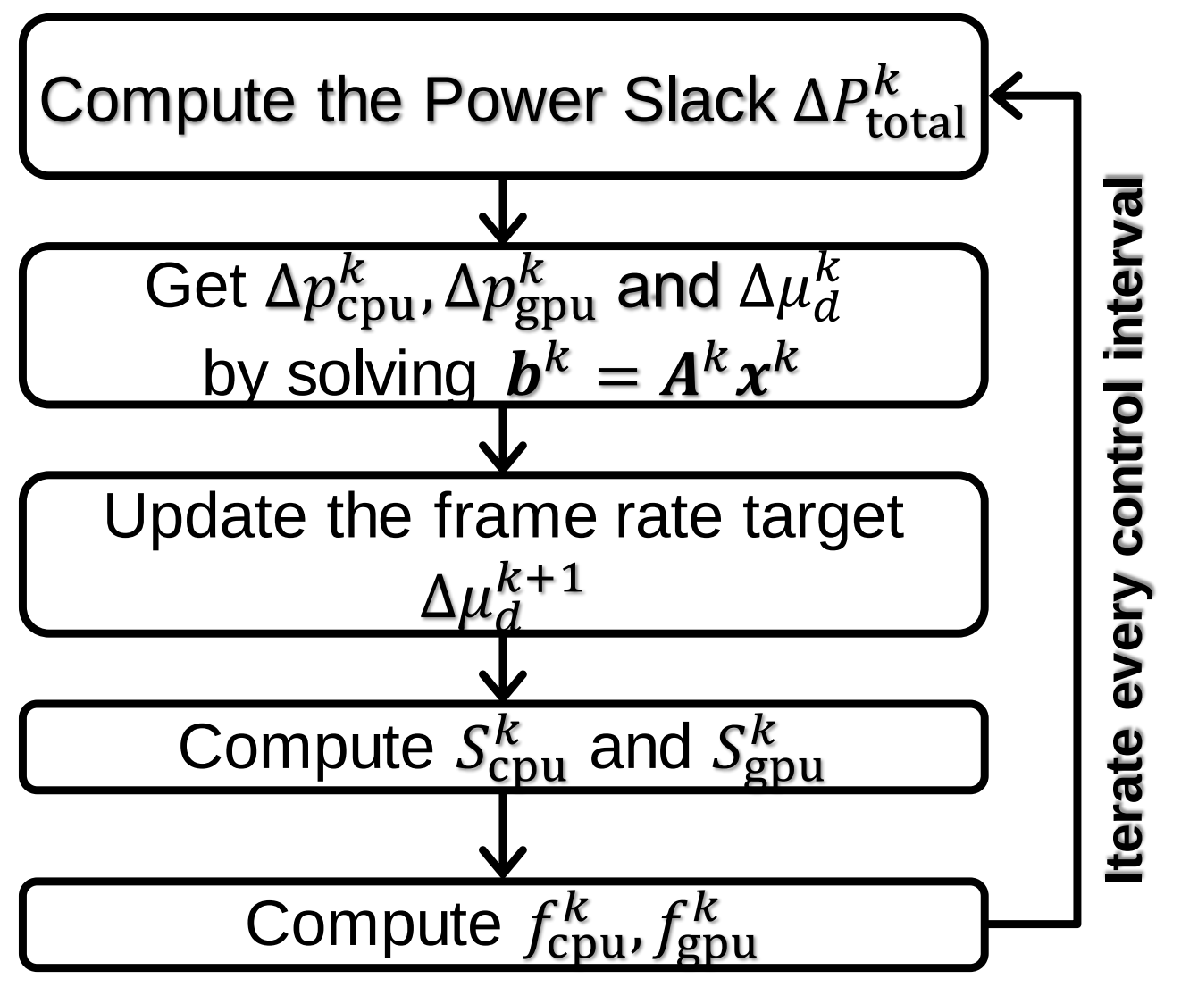


Fig. 9: CPU-GPU Work In Tandem



Compute the Power Slack $\Delta P_{\text{total}}^k$

Get $\Delta p_{\text{cpu}}^k, \Delta p_{\text{gpu}}^k$ and $\Delta \mu_d^k$ by solving $\boldsymbol{b}^k = \boldsymbol{A}^k \boldsymbol{x}^k$

Update the frame rate target $\Delta \mu_d^{k+1}$

Compute S_{cpu}^k and S_{gpu}^k

Compute $f_{\text{cpu}}^k, f_{\text{gpu}}^k$

Iterate every control interval

Fig. 10:Example Control Algorithm

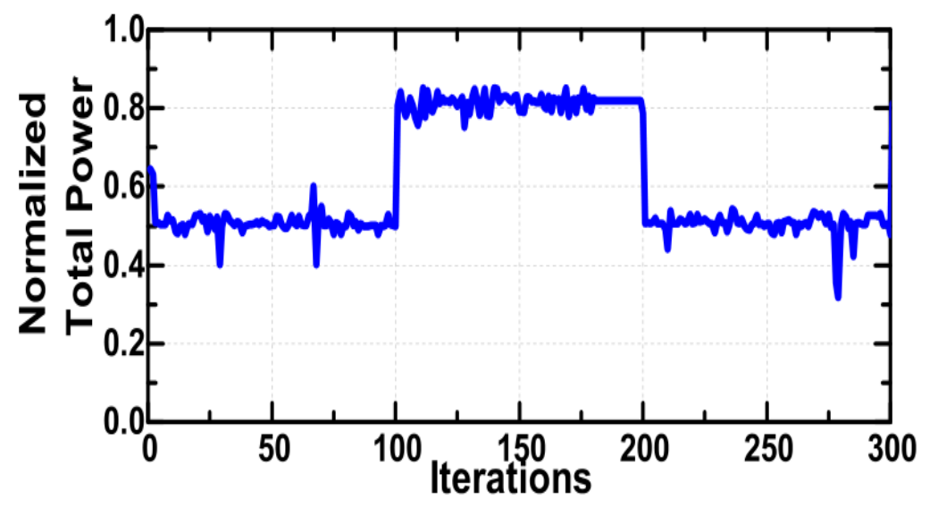


Fig. 11: Total Power changes with number of iterations.

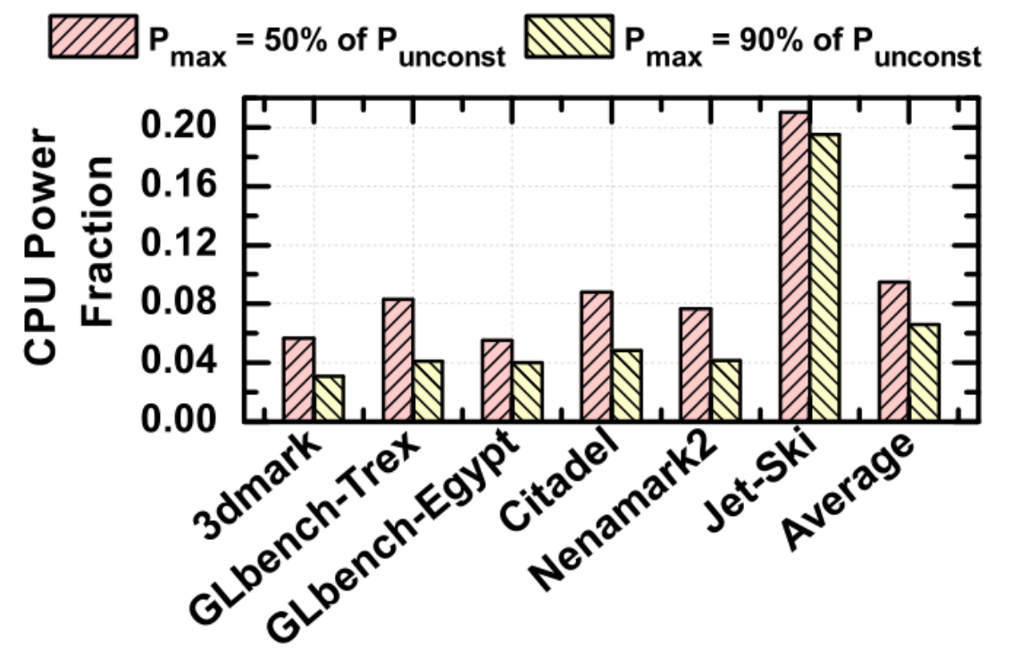


Fig. 12: CPU power changes as we choose different P_{max}.

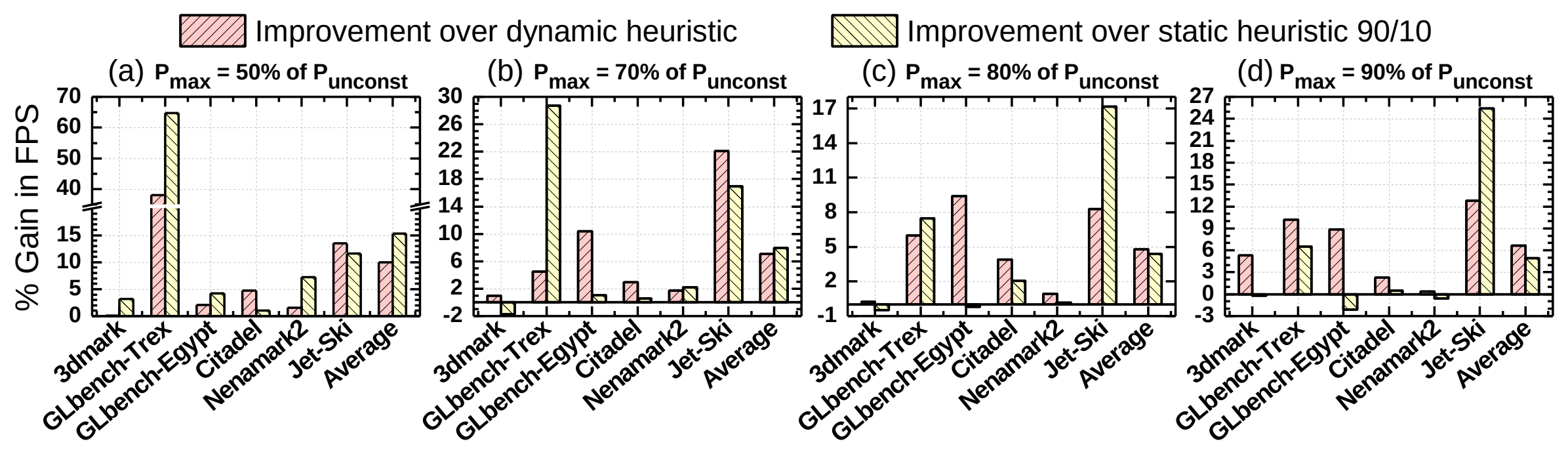


Fig. 13: Comparison of FPS (frame rates) for our approach, Heuristic-dynamic, and Heuristic-static-90/10

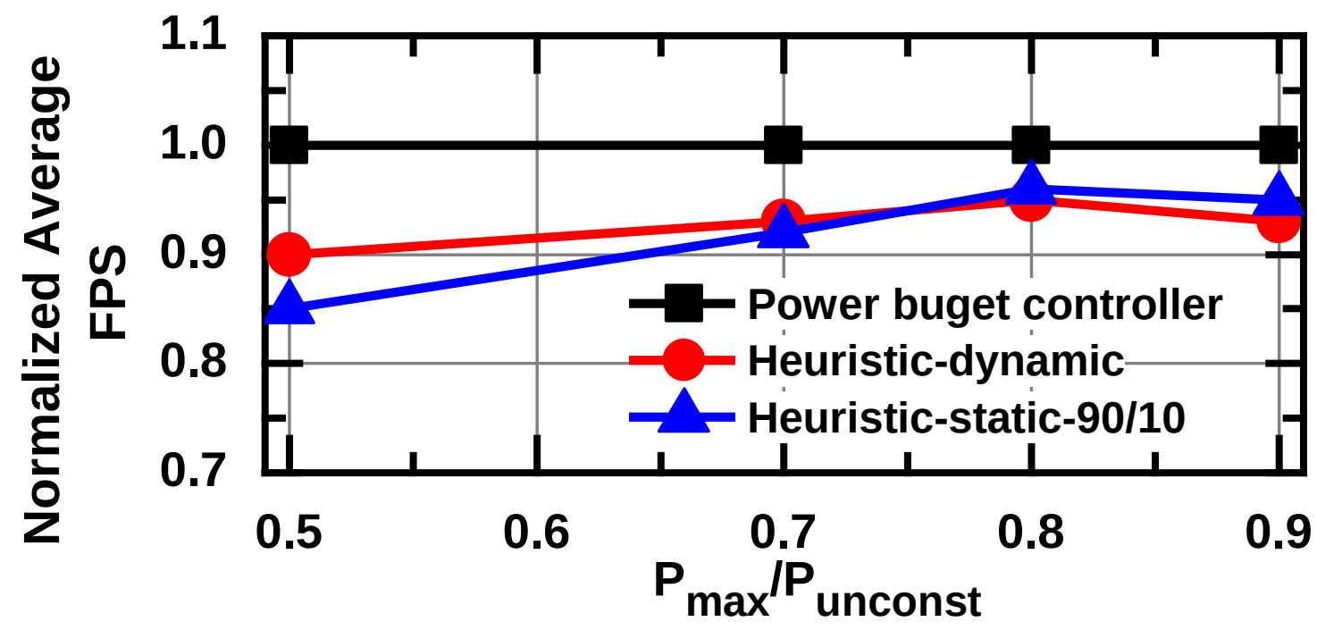


Fig. 14: Comparison of average FPS (frame rates) for our approach, Heuristic-dynamic, and Heuristic-static-90/10

Experimental Results

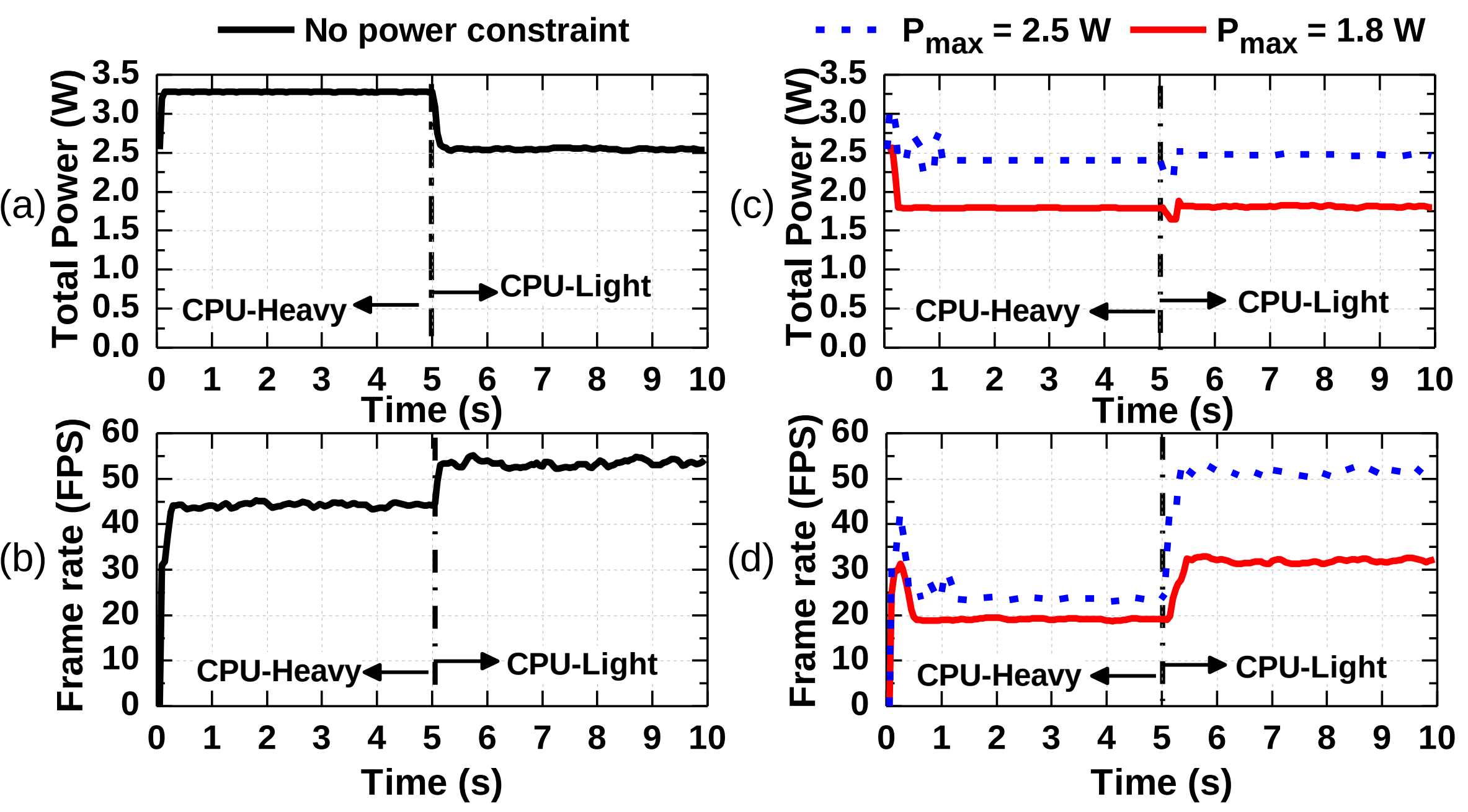


Fig. 15: Total Power and FPS when there is no power constraint (a) and (b); and when there is power constraint of P_{max} (c) and (d).

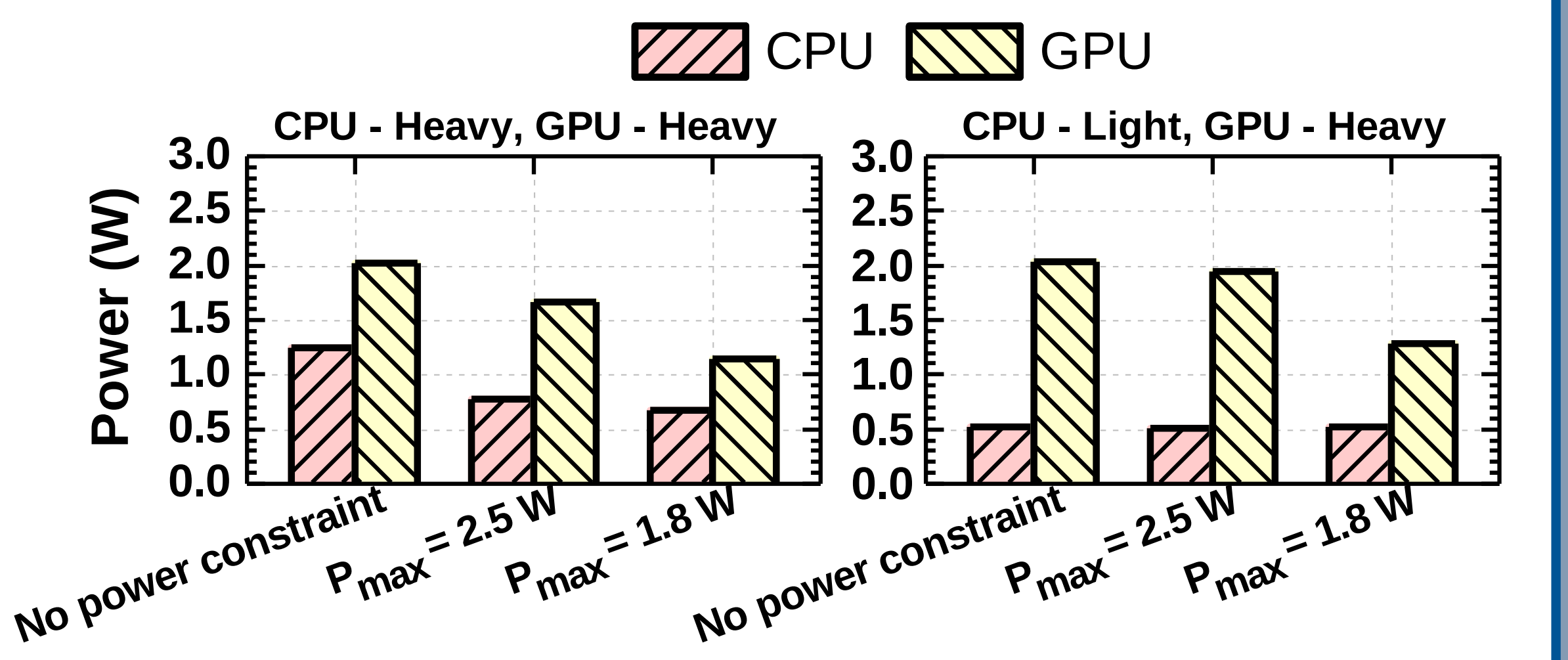


Fig. 16: Power consumption when the application is CPU-Heavy, GPU-Heavy in (a); and when the application is CPU-Light, GPU-Heavy in (b).

- It is critical to distribute power budget efficiently between the CPU and GPU.
- Our technique provides high throughput by efficient distribution of power slack.
- Results using mobile platform show average increase of 15% in frame rate compared to existing algorithms.

Conclusion and Future Directions

- Developed a dynamic reconfiguration framework for mulicore SoCs, and demonstrated its feasibility through dynamic cache reconfiguration as well as dynamic power budgeting.
- Our future work includes developing techniques for NoC reconfiguration as well as exploring synergy between computation, communication and storage reconfiguration under energy, thermal, reliability and real-time constraints.

References

- Y. Huang and P. Mishra, Reliability & Energy-aware Cache Reconfiguration for Embedded Systems, IEEE Intl. Symp. on Quality Electronic Design (ISQED), 2016. **Best Paper Award**
- M. Chen, X Zhang, G. Pu, X. Fu, P. Mishra, Resource Constrained Scheduling using Parallel Structure-Aware Pruning Techniques, IEEE Trans. on Computers, 65(7), 2016.
- Y. Huang and P. Mishra. Vulnerability-aware Energy Optimization using Reconfigurable Caches in Multicore Systems. Submitted to ICCAD 2017.
- Z. Wang, S. Ranka, P. Mishra, Task Partitioning and Scheduling for Thermal Management in Multicore Processors, IEEE Intl. Symposium on Quality Electronic Design (ISQED), 2015.
- M. Chen, D. Yue, X. Qin, X. Fu and P. Mishra, Variation-Aware Evaluation of MPSoC Task Allocation and Scheduling Strategies using Statistical Model Checking, DATE 2015.
- G. Singla, G. Kaur, A. K. Unver and U. Y. Ogras. Predictive dynamic thermal and power management for heterogeneous mobile platforms. DATE 2015.
- U. Gupta, R. Ayoub, M. Kishinevsky, D. Kadjo, N. Soundararajan, U. Tursun and U. Y. Ogras. Dynamic power budgeting for mobile systems running graphics workloads, IEEE Transactions on Multiscale Computing Systems, 2017.

Acknowledgments: This work was partially supported by the NSF grant (CNS-1526687) and Intel Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of the sponsors.