CAPE: Criticality-Aware Performance and Energy Optimization Policy for NCFET-based Caches

DIVYA PRANEETHA RAVIPATI, IIT Delhi
RAMANUJ GOEL, IIT Delhi
VICTOR M. VAN SANTEN, Technical University of Munich
HUSSAM AMROUCH, Technical University of Munich
PREETI RANJAN PANDA, IIT Delhi

1 APPENDIX A

1.1 FinFET-based LLC results evaluation

We evaluated the *CAPE* policy for FinFET-based LLC along with the baselines. V_top is set to V_{max} to apply *CAPE* for FinFET-based caches. Figure 1 shows the cache energy savings, slowdown, EDP savings, and throughput degradation with the baseline policies and CAPE compared to operating the LLC at 0.7 V. From our experimental evaluations, we observe that operating the FinFET-based cache at 0.3 V (V_{min}) resulted in higher energy and EDP savings while degrading throughput. However, CAPE chooses to operate at a voltage (may not be V_{min}) to reduce energy with minimal impact performance. We observe that CAPE results in the energy and EDP savings of 60 % similar to the best performing baseline policies for FinFET-based LLC along with a similar slowdown and throughput degradation of 3.8 %.

1.2 Sensitivity analysis of the thresholds

The criticality threshold (*criticality_th*) and interval threshold (*interval_th*) have to be identified for the *CAPE* policy. We have attempted to experimentally determine the thresholds that cause the performance to be degraded by no more than 5%. From our experimental analysis, we observed that performance is impacted negatively with increasing *criticality_th*, as *CAPE* does not consider increasing the LLC voltage until the fraction of time for which the LLC blocks the instruction commit exceeds the *criticality_th* (the instructions are blocked for a longer duration and the cache voltage is not modified until a higher *criticality_th*, thus degrading performance). Considering a worst-case performance degradation of 5 %, *criticality_th* is set to 20 % for *CAPE_sim*. For similar performance degradation of 5 %, *criticality_th* is set to 30 % and M_{Th} is set to 40% for *CAPE* which uses computed *CCF*.

Interval threshold (*interval_th*) helps in making policy decisions by observing the past behavior. Having too high *interval_th* leads to decisions that do not suit the present application phase behavior. Setting *interval_th* too low ignores the recent application phase behavior, resulting in performance degradation and frequent voltage switching. With *interval_th* set to 20, the maximum voltage switchings across all the benchmark mixes are within 16 % of the total intervals, and the performance degradation is within 5%.

We also need to identify the interval (epoch) size after which the policy has to be invoked. We varied the interval size from 50 µs till 500 µs in steps of 100 µs (from 100 µs interval size onward). Increasing interval sizes lead to degraded performance, as changes in application behavior are missed and are not reacted to with the appropriate voltage. Smaller interval sizes lead to frequent invocation of the policy and voltage changes which have energy overhead (e.g., charging and

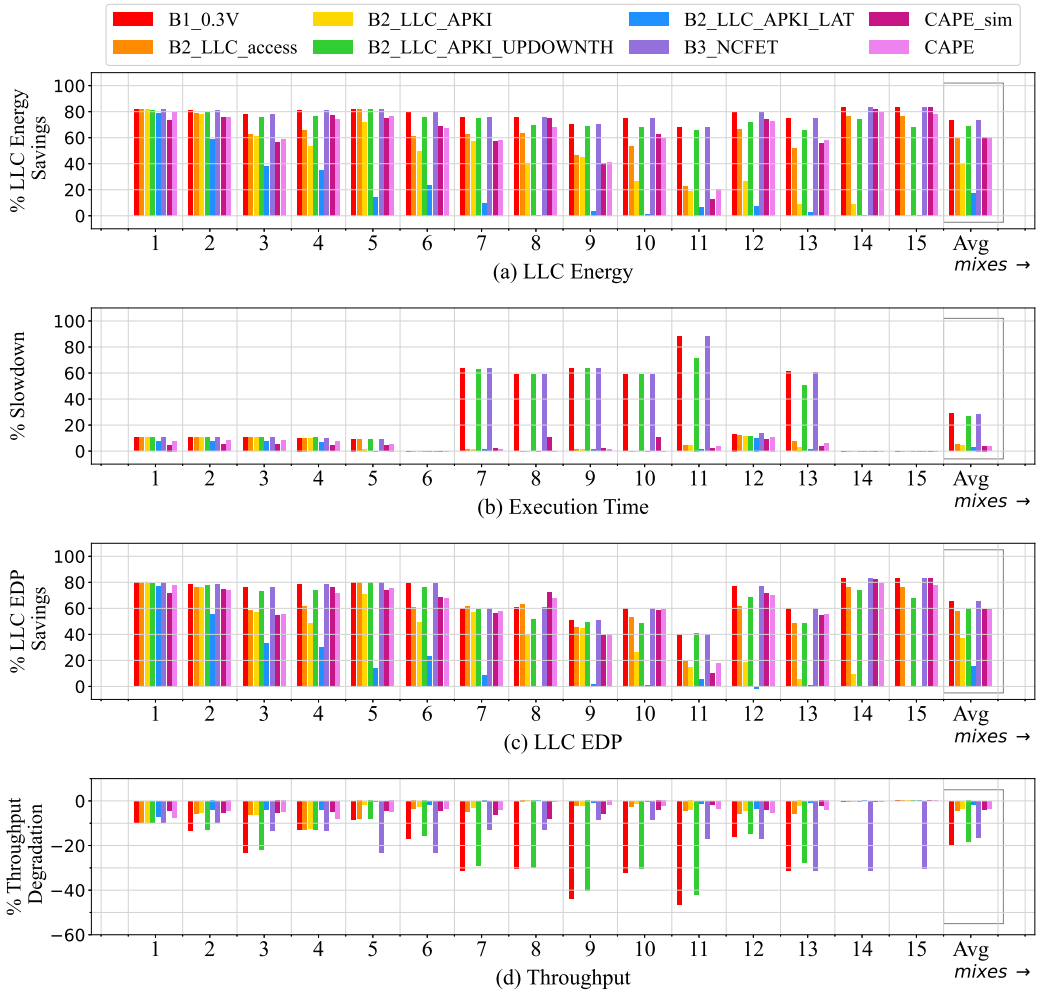
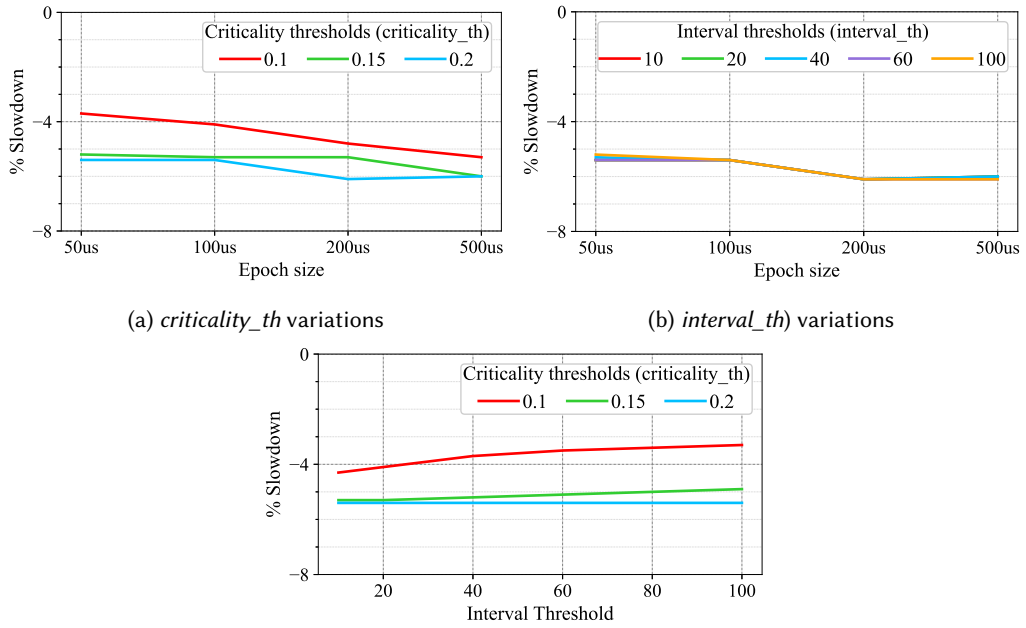


Fig. 1. Normalized FinFET-based LLC Energy savings, Slowdown, EDP savings, and Throughput degradation with the baseline policies and CAPE in percentages compared to operating the LLC at 0.7 V

discharging the circuit to higher/lower voltages). We use an interval size of 100 µs in our simulations and assume an overhead of 100 ns for each voltage step change [1].

We have evaluated results while operating CAPE using various thresholds. Figure 2 shows the impact on the percentage slowdown at various thresholds compared to operating LLC at Vmax (0.7V). We try to decide the thresholds at which the worst case performance degradation does not exceed 5%. Setting *criticality_th* to 20%, interval*interval_th* to 20, and interval size of 100 µs, the worst case performance degradation observed is 5.2% which we used in our paper. We can operate at lower *criticality_th* as well for reduced performance degradation, but we have tried the higher bound so that the policy takes decisions at a higher cache criticality level and still result in improved LLC energy and EDP savings. We also observe that the worst-case slowdown through any combination of the thresholds is not beyond 6%, implying the robustness of the policy against variations in thresholds.



(a) *criticality_th* variations

(b) *interval_th*) variations

(c) *criticality_th* and *interval_th*) variations

Fig. 2. Sensitivity analysis of the thresholds (a) Variation in percentage slowdown at various interval sizes and criticality thresholds (*criticality_th*) while fixing interval threshold (*interval_th*) to 20, (b) Variation in percentage slowdown at various interval sizes and interval thresholds (*interval_th*) while fixing criticality threshold (*criticality_th*) to 0.2, and (c) Variation in percentage slowdown at various interval thresholds and criticality thresholds while fixing interval size to 100 µs.

1.3 Evaluating CAPE with a few more conventional measurements

We compare the *CAPE* policy against seven baseline policies described below. Five of the considered baseline policies are adapted from traditional CMOS(/FinFET)-based system energy optimization techniques, which rely on activity metrics [3, 4] and are NCFET-unaware. Another baseline policy is based on state-of-the-art energy optimization work for NCFET-based system [2].

1.3.1 Operating at the lowest possible voltage to save power (B1_Vmin). One simple yet traditional cache power-saving technique includes running the cache at the lowest operating voltage (V_{min}). However, operating at the lowest voltage does not guarantee energy savings, as performance could be degraded.

1.3.2 Conventional energy saving policies based on activity. To improve performance while also striving to save power, traditional policies depend on the activity in the system [3]. We implement the following baseline policies for the metrics that are identified to represent cache activity.

B2_Access (Based on accesses) & B2_APKI (Based on APKI). Cache accesses and APKI metrics capture the activity in the cache. If the cache activity defined by the metrics exceeds certain thresholds, then the appropriate voltage is applied to the cache.

B2_Delta_APKI (Based on ΔAPKI). We also considered a differential metric which captures the difference in APKI in successive intervals ($\Delta APKI$). If $\Delta APKI$ exceeds an upper threshold, then the

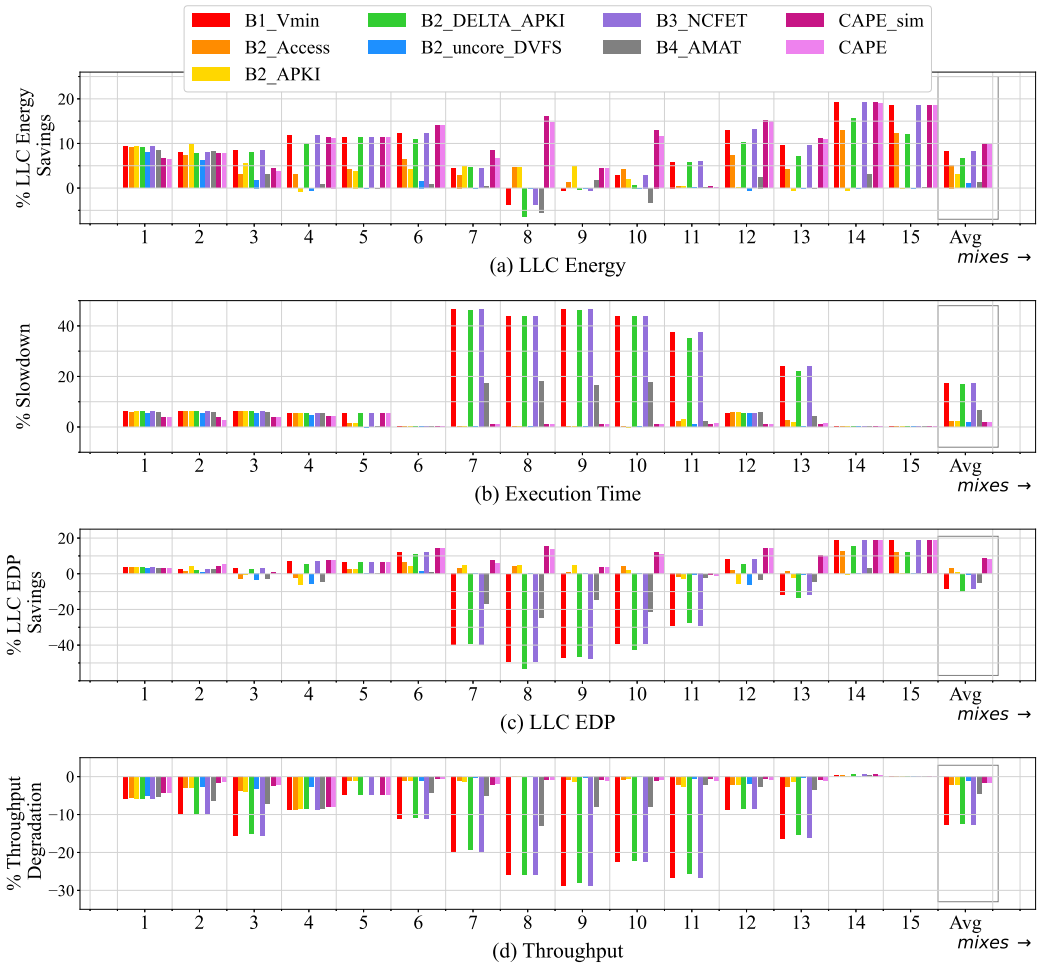


Fig. 3. Normalized LLC Energy savings, Slowdown, EDP savings, and Throughput degradation with the baseline policies and CAPE in percentages compared to operating the LLC at 0.7 V

policy considers increasing the cache voltage by one step. If $\Delta APKI$ is less than a lower threshold, then the voltage is decremented by one step. Otherwise, the voltage remains the same.

B2_uncore_DVFS: Based on the uncore DVFS policy proposed by Won et al. [4]. We adapted the uncore DVFS policy presented by Won et al. [4], as the core operates at a fixed voltage and frequency in our work. The uncore voltage decisions are based on average uncore latency and injection rate (APKI), which leads to applying higher voltages for the application phases with high miss rates due to higher APKI.

1.3.3 NCFET-aware energy optimization policy (B3_NCFET). State-of-the-art work on NCFET-aware energy optimization policy [2] evaluates all the available voltages and applicable frequencies to minimize energy every interval. Since frequency is fixed in our work, we adapt the baseline to iterate through all the operating voltages and choose the voltage that is expected to cost the least energy for the interval.

Table 1. SPEC CPU2017 benchmark mixes - Test set

mix	Benchmark mixes running on 8 cores
1	x264, exchange2, leela, wrf, imagick, nab, perlbench, xz
2	nab, gcc, leela, xz, xalancbmk, lbm, xalancbmk, lbm
3	xalancbmk, lbm, mcf, wrf, xalancbmk, lbm, mcf, wrf
4	x264, imagick, nab, perlbench, exchange2, lbm, leela, omnet
5	exchange2, perlbench, xalancbmk, povray, imagick, omnet, lbm, leela

1.3.4 B4_AMAT: Based on Average memory access latency . Average memory access latency (AMAT) represents the round trip time starting from the time the memory request is made, until the time the request is served. The entire request path may constitute traversing through LLC and main memory. Since AMAT calculation also includes main memory access latency, the computed AMAT value will be high whenever LLC miss rate is high, leading to policy taking the voltage decision accordingly. However, in such cases, LLC accesses are not the bottleneck for performance, since the accesses are currently to main memory.

1.3.5 CAPE. We evaluate the *CAPE* policy using the computed *CCF* value to verify how close the results are to those obtained with accurate *CCF* calculation. We call the policy using the accurate *CCF* obtained from Sniper simulator as ***CAPE_sim*** and the policy using the computed *CCF* as ***CAPE***.

Figure 3 shows the evaluation of CAPE policies with the baselines for LLC energy savings, slowdown, LLC EDP savings and throughput degradation. In this article, we focus on the comparing the results with *B4_AMAT*. Since AMAT captures round trip latency of a memory access request, the computed AMAT value will be high whenever there are more LLC misses, due to higher off-chip memory access latency. Thus, the policy decides to increase the LLC voltage. However, increasing LLC voltage in such cases will not improve performance and also results in LLC energy consumption. We observe from Figure 3 that for mixes (mix_13, mix_15) where LLC accesses are high and also miss rate is high, *B4_AMAT* applies higher voltages, resulting in reduced LLC energy savings with no improvement in performance. Our experimental evaluations show that *B4_AMAT* policy does not result in better LLC energy/EDP savings.

1.4 Evaluating adaptability of CAPE to new workloads (Test set)

To ensure the policy's robustness against any new set of workloads, we deliberately selected a mix of workloads that had not been previously tested with the policy (new set of SPEC CPU2017 benchmarks such as `wrf`, `mcf`, `xalancbmk`, and `povray` are used in the test set). This approach ensures thorough testing across a wider range of scenarios, enhancing the policy's adaptability and effectiveness. Table 1 shows the set of workload mixes considered for the test set.

Figure 4 shows the LLC energy savings, execution time, LLC EDP savings and throughput degradation with the baseline policies and CAPE compared to operating the LLC at 0.7V. Figure 4 shows that the CAPE policy results in higher performance (smaller execution times) and results in improved LLC EDP savings compared to that of the baselines.

1.5 Adaptability of CAPE to different cache sizes and system configurations

We have conducted experiments on the following cache and system configurations, apart from the system configuration described in the manuscript (8 cores, 16 MB LLC). (i) 8 cores, 8 MB LLC (1MB LLC slice per core), (ii) 4 cores, 4 MB LLC (1MB LLC slice per core), and (iii) 4 cores, 8 MB LLC (2MB

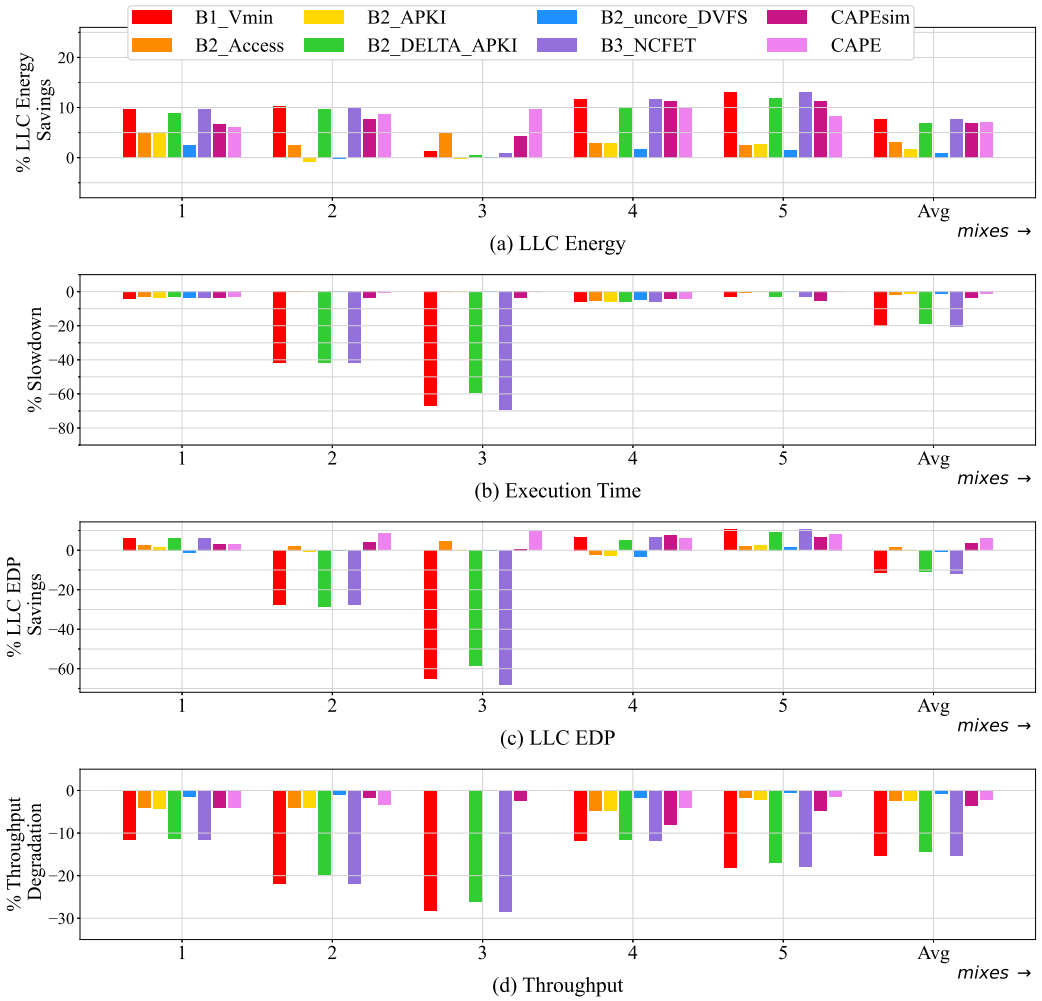


Fig. 4. Normalized LLC Energy savings, Slowdown, EDP savings, and Throughput degradation with the baseline policies and CAPE in percentages compared to operating the LLC at 0.7 V on a new set of workloads

LLC slice per core) We have repeated our experiments on all the six baseline policies and two CAPE policies (*CAPE_sim* and *CAPE*) for the above mentioned three system configurations. Figures 5, 6, and 7 show the LLC energy savings, slowdown, LLC EDP savings and throughput degradation with the baselines policies and CAPE compared to operating at $Vmax$ (0.7 V) for various system configurations. Though a few baseline policies show better LLC energy savings compared to CAPE, CAPE yields reduced slowdown performance, resulting in overall better LLC EDP savings. We observe that the thresholds are robust to various cache sizes and system configurations, since the thresholds (*criticality_th*, Mth) in our case are relative values.

REFERENCES

[1] Stijn Eyerman and Lieven Eeckhout. Fine-grained dvfs using on-chip regulators. *ACM Trans. Archit. Code Optim.*, 8(1), feb 2011. ISSN 1544-3566. doi: 10.1145/1952998.1952999. URL https://doi.org/10.1145/1952998.1952999.

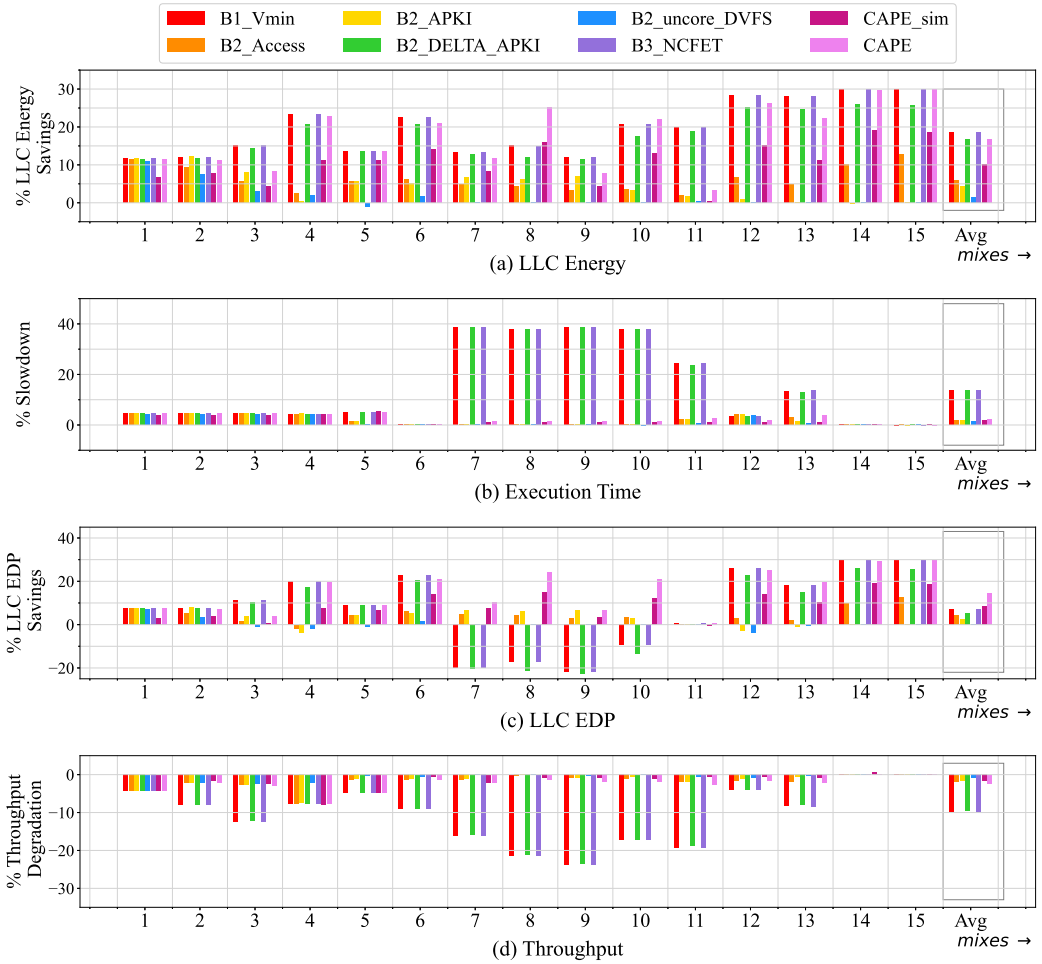


Fig. 5. Normalized LLC Energy savings, Slowdown, EDP savings, and Throughput degradation with the baseline policies and CAPE in percentages compared to operating the LLC at 0.7 V on 8 cores, 8 MB LLC system configuration

[2] Sami Salamin, Martin Rapp, Jörg Henkel, Andreas Gerstlauer, and Hussam Amrouch. Dynamic power and energy management for NCFET-based processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3361–3372, 2020. doi: 10.1109/TCAD.2020.3012644.

[3] E. Talpes and D. Marculescu. Toward a multiple clock/voltage island design style for power-aware processors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(5):591–603, 2005. doi: 10.1109/TVLSI.2005.844305.

[4] Jae-Yeon Won, Paul V. Gratz, Srinivas Shakkottai, and Jiang Hu. Resource sharing centric dynamic voltage and frequency scaling for cmp cores, uncore, and memory. *ACM Trans. Des. Autom. Electron. Syst.*, 21(4), may 2016. ISSN 1084-4309. doi: 10.1145/2897394. URL https://doi.org/10.1145/2897394.

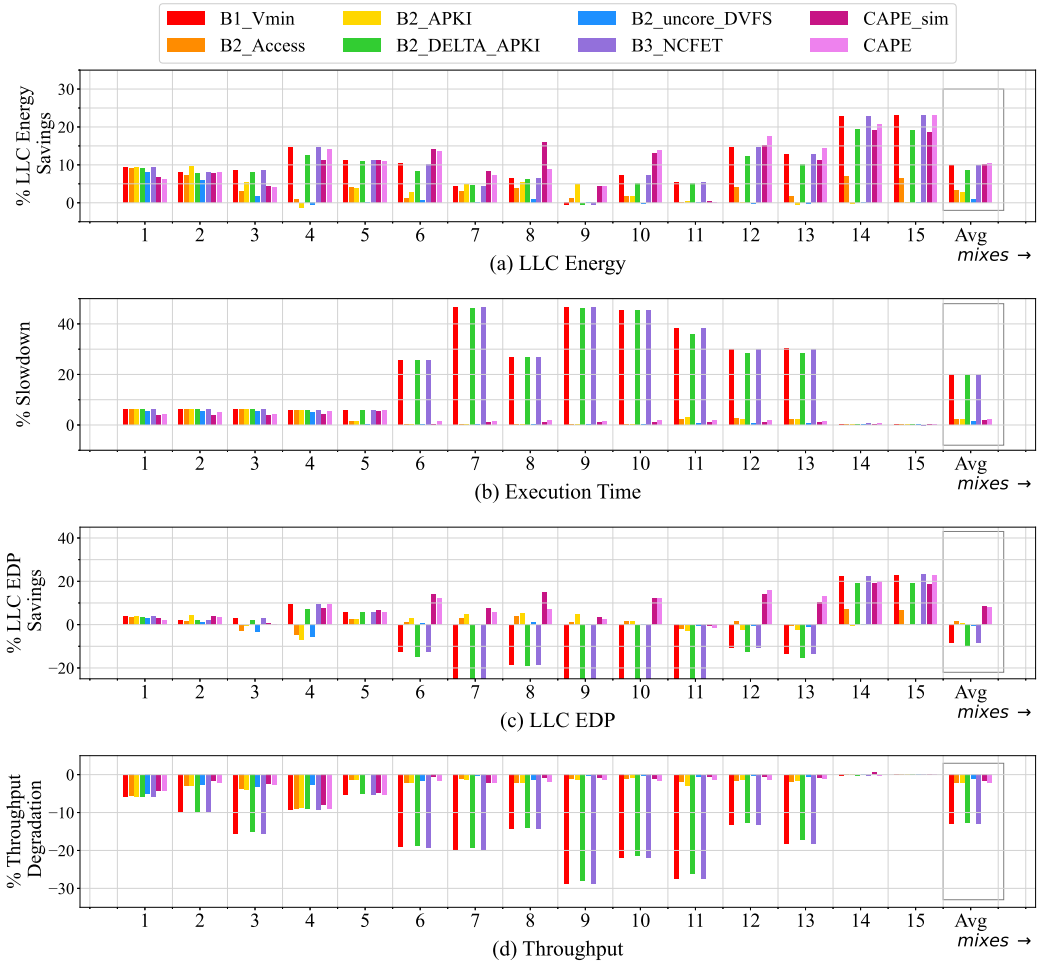


Fig. 6. Normalized LLC Energy savings, Slowdown, EDP savings, and Throughput degradation with the baseline policies and CAPE in percentages compared to operating the LLC at 0.7 V on 4 cores, 8 MB LLC system configuration

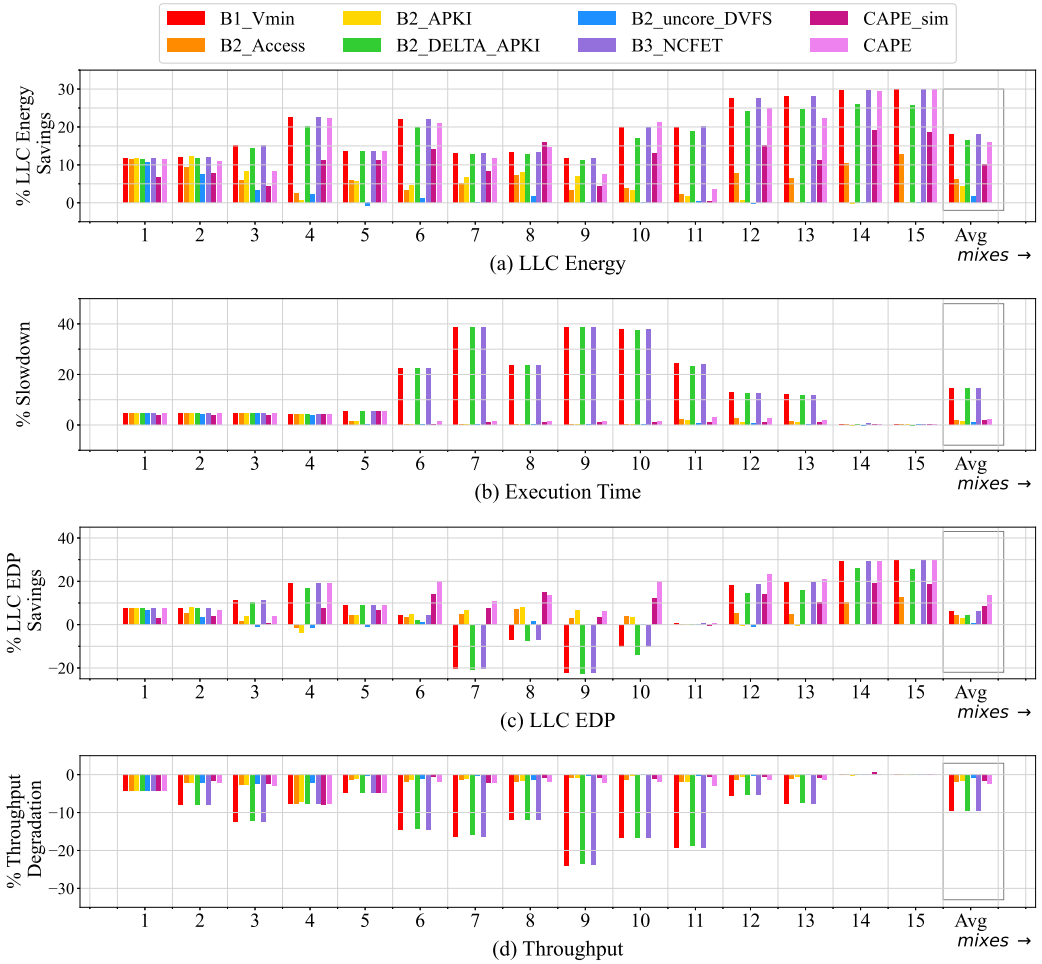


Fig. 7. Normalized FinFET-based LLC Energy savings, Slowdown, EDP savings, and Throughput degradation with the baseline policies and CAPE in percentages compared to operating the LLC at 0.7 V on 4 cores, 4 MB LLC system configuration