An Evaluation of Per-Chip Non-Uniform Frequency Scaling on Multicores

Xiao Zhang

Kai Shen

Sandhya Dwarkadas

Rongrong Zhong

USENIX ATC 2010   06/25/2010

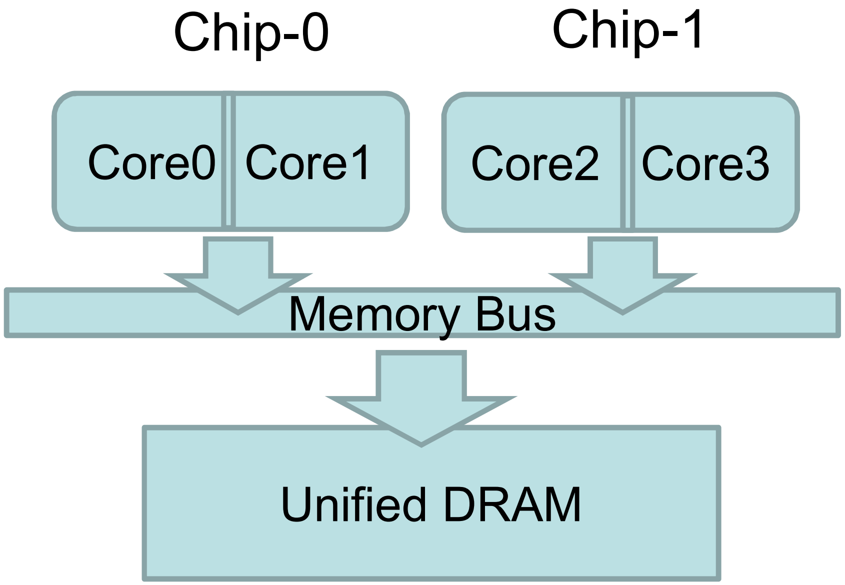
Dynamic Voltage/Frequency Scaling (DVFS) on Multicore Chips

- Efficient for memory intensive applications
  - Significant CPU power savings with no (or little) performance loss

- Current constraint: a single voltage setting applies to all sibling cores
  - E.g., Intel and AMD processors
  - Limits power savings opportunities if memory intensive and non-intensive applications run on the same chip

Targeted Multicore Platforms

- Multichip machines have opportunities for per-chip non-uniform voltage/frequency settings
- Symmetric Multiprocessing (SMP) based multi-chip multicore machines



Outline

- A smart scheduling to facilitate per-chip frequency scaling for power savings (with competitive/better performance)

- Frequency-to-performance model for flexible power management

Similarity Grouping Scheduling

- Group applications with similar cache miss ratio on the same chip
  - Separate high and low miss ratio applications on different chips
  - High-miss-ratio chip running at low frequency while low-miss-ratio chip running at high frequency
- Additional benefits on addressing resource contention
  - Mitigate cache thrashing effect
  - Avoid over-saturating memory bandwidth

Evaluation Setup

- Platform
  - 2-chip Intel 3GHz WoodCrest processor (two cores per chip, sharing 4MB L2 cache) SMP running Linux-2.6.18
  - Frequency at 3 / 2.67 / 2.33 / 2 GHz via writing Intel-specific IA32_PERF_CTL registers

- Overall performance = $\sqrt[n]{P_1 * P_2 * \cdots * P_n}$
  (geometric mean of running applications' performance)

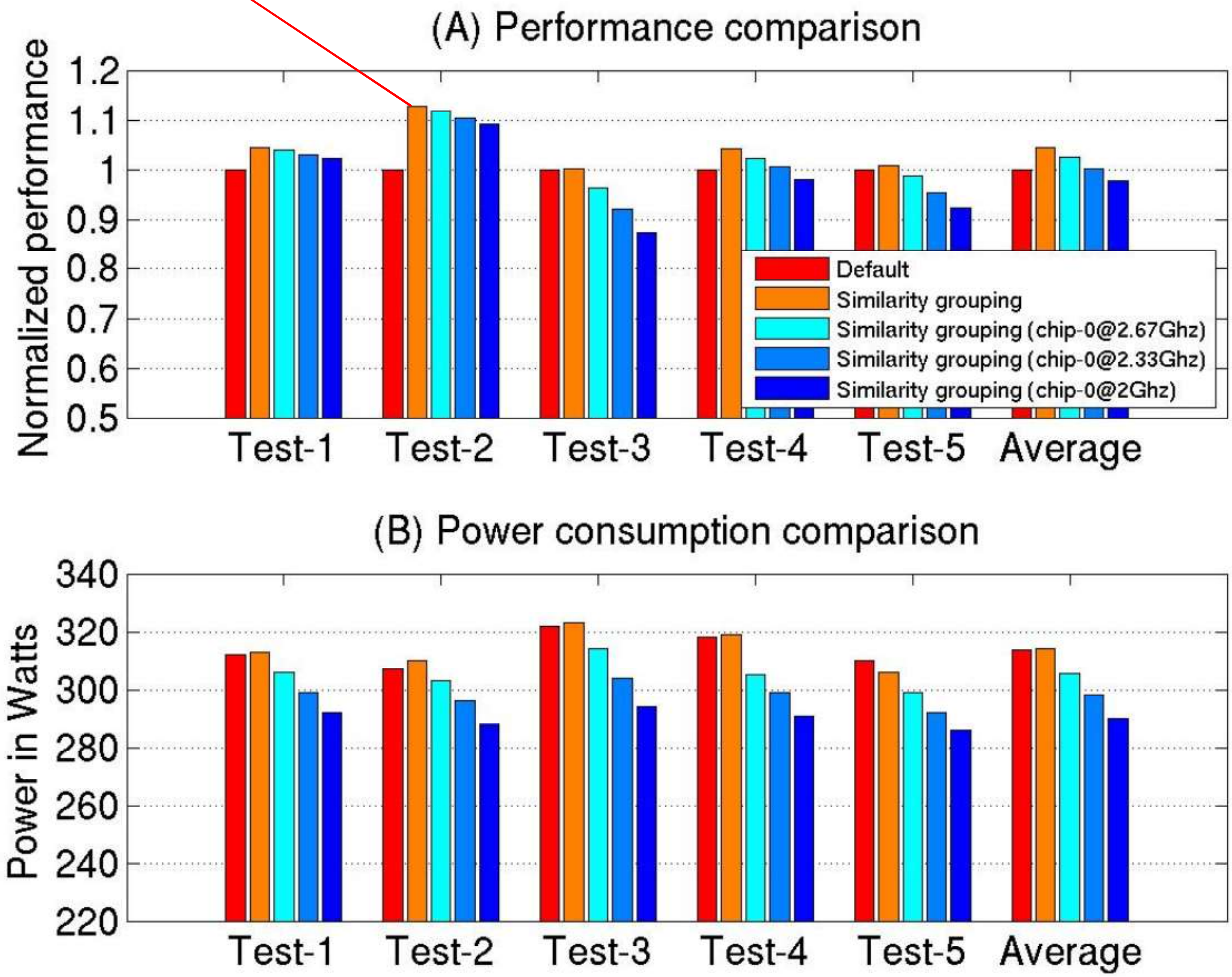
Evaluation Setup

- Benchmarks
  - 12 SPECCPU 2000 applications and 2 server-style applications divided into 5 test sets

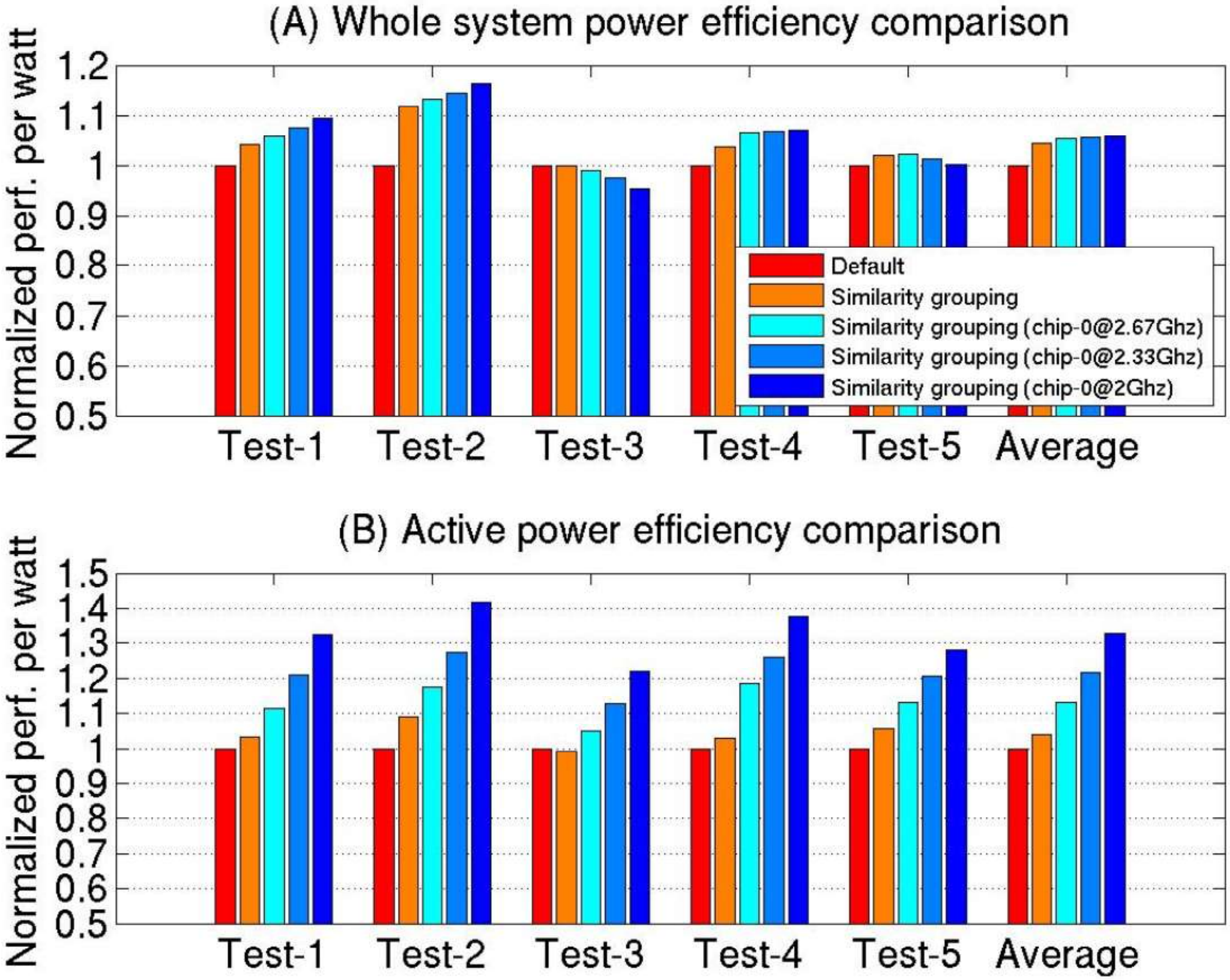
Similarity Grouping	Chip-0 (high miss ratio)	Chip-1 (low miss ratio)
Test #1	{equake, swim}	{parser, bzip}
Test #2	{mcf, applu}	{art, twolf}
Test #3	{wupwise, mgrid}	{mesa, gzip}
Test #4	{mcf, swim, equake, applu, wupwise, mgrid}	{parser, gzip, bzip, mesa, twolf, art}
Test #5	Two SPECjbb threads	Two TPC-H threads

Static Frequency Scaling

Avg. 25% reduction in cache misses



Power Efficiency (Performance per Watt)


(A) Whole system power efficiency comparison

(B) Active power efficiency comparison

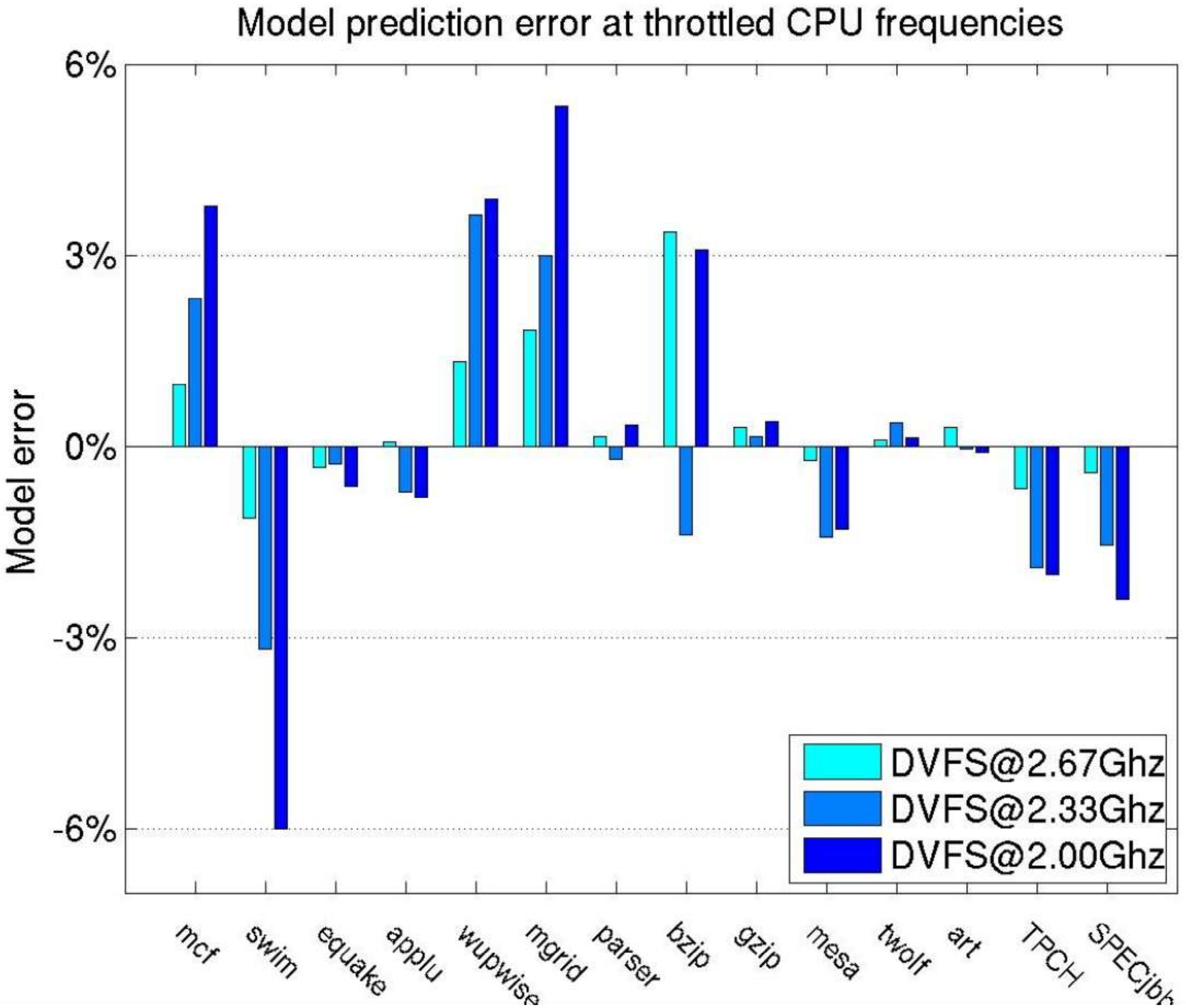
Frequency-to-Performance Model

- Objective: explore power savings with bounded performance loss
- Assumptions
  - An application’s performance is linearly determined by cache and memory access latencies
  - Frequency scaling only affects on-chip accesses
  - Miss ratio does not vary across frequencies

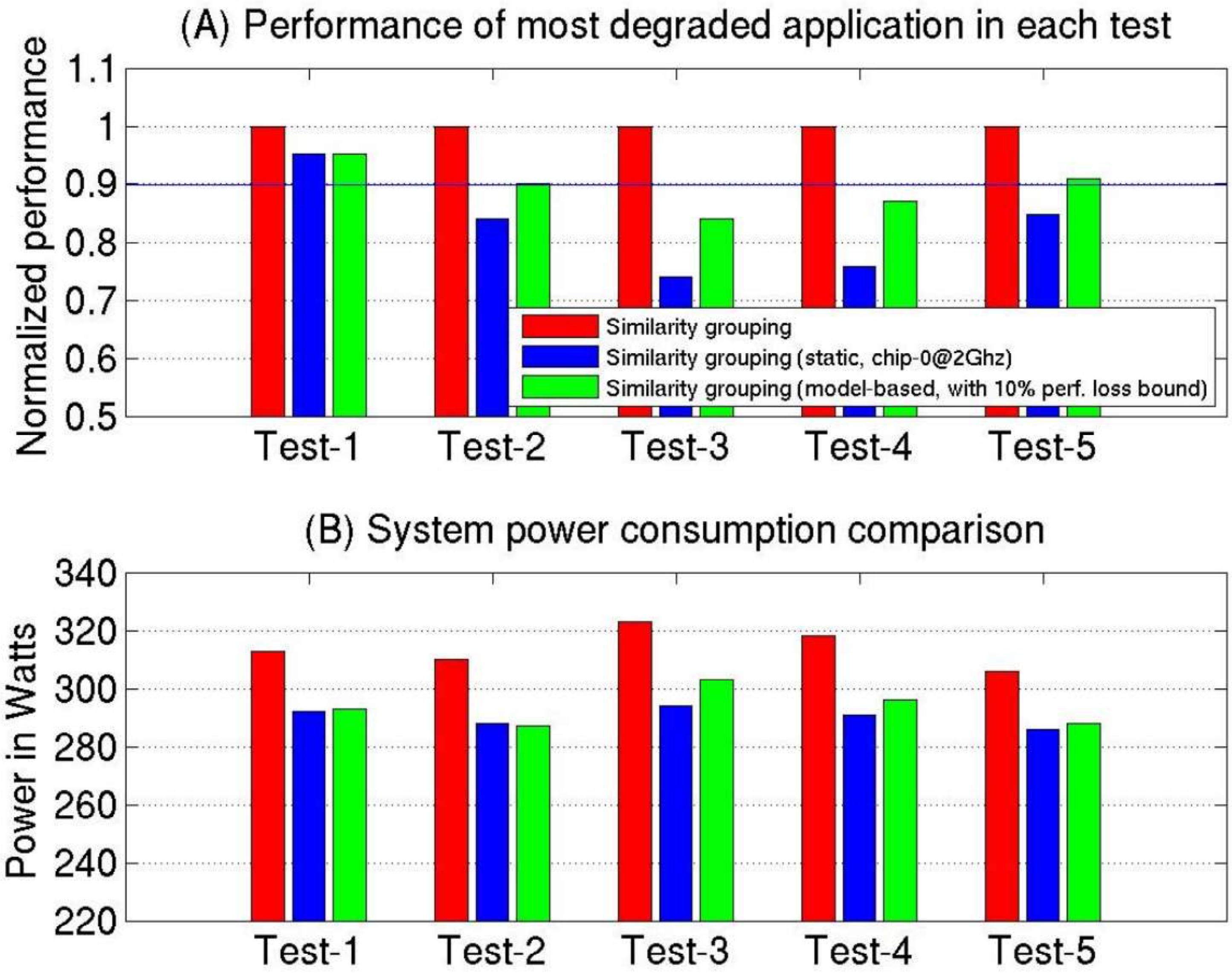
$$T(f) \approx \frac{F}{f} * HitRatio * L_{CacheHit} + MissRatio * L_{CacheMiss}$$

Normalized performance at frequency f = $T(F) / T(f)$

Model Accuracy

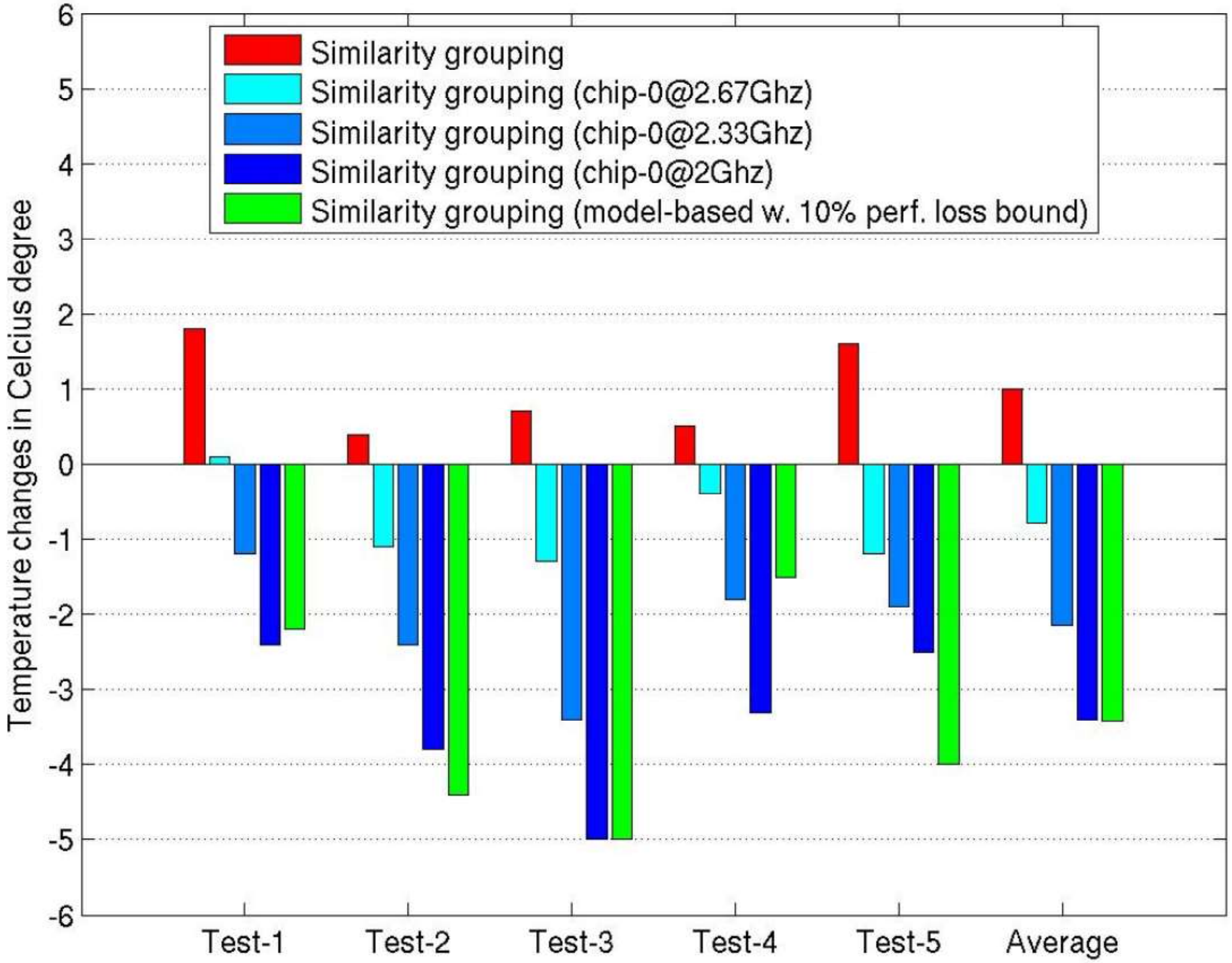


Model-based Dynamic Frequency Setting


(A) Performance of most degraded application in each test

(B) System power consumption comparison

Thermal Reduction over Default System



Summary

- Similarity grouping Improves performance due to reduced resource contention and facilitates per-chip frequency scaling for power savings
- Guided by a simple frequency-performance model, we achieve ~20 watts power savings and ~3 Celsius degrees CPU thermal reduction with bounded performance loss