Analysis of Cross-layer Vulnerability to Dynamic Variations: An Adaptive Instruction-level to Task-level Approach

Abbas Rahimi†, Luca Benini‡ and Rajesh Gupta†
†UC San Diego and ‡Università di Bologna

Variation-aware Cross-layer Vision
* We introduce the notions of ILV, SLV, PLV, and TLV to expose variation and its effects to the software stack for use in compiler and runtime optimizations.
* Effect of voltage and temperature variations on the performance of a tightly-coupled shared-L1 processor cluster in 45nm TSMC technology

Non-uniform Instruction-level Vulnerability (ILV)
* Instructions have different levels of vulnerability to variations depending on the way they exercise the critical paths across the various pipeline stages.
* Our analysis shows that ILV is not uniform across SPARC V8 instruction set.
* ILV data partitions instructions into three classes [DATE’12]:
  (i) logical and arithmetic, (ii) memory, and (iii) HW multiply and divide
* For every operating conditions: ILV (3rd class) ≥ ILV (2nd class) ≥ ILV (1st class)

Table shows inter-core ILV of three classes of instructions; ILV = Number of violated cycles over the total number cycles during Monte Carlo simulation

Task-level Vulnerability (TLV)
* Hardware supports variability monitors with online per-core characterization of TLV metadata (decentralized TLV Characterization)
* Software supports a Task-level Errant Instruction Management technique to utilize TLV metadata in the runtime task scheduler.
* IPC of a 16-core processor cluster is increased up to 1.51× (1.17× on average). We evaluate the effectiveness of our approach across a wide temperature range (∆T=90°C).

---

**Additional Information**