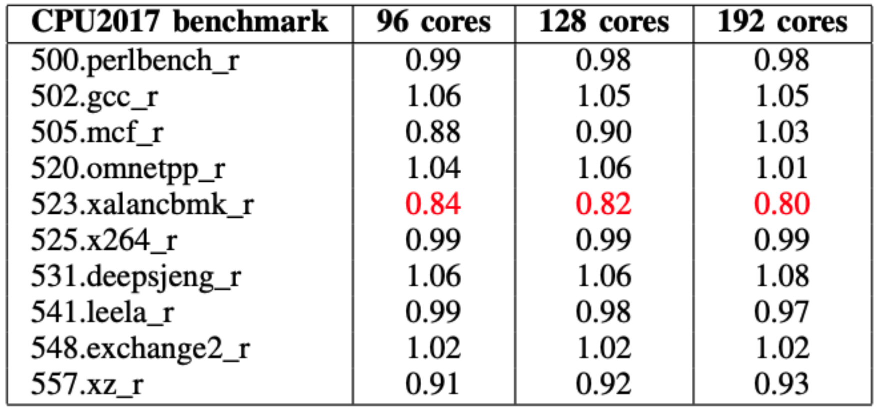
Memory Access Vectors: Improving Sampling Fidelity for CPU Performance Simulations

Sriyash Caculo, Mahesh Madhav, Jeff Baxter
AmpereOne Product Architecture
Ampere Computing, Portland, OR

Challenges in SoC Performance Projections

- **Performance projection** accuracy is critical for CPU architects
  - Run benchmarks in software models to forecast performance
- **SimPoint sampling** with Basic Block Vectors (BBVs) is widely adopted
  - Software simulation is slow -> sampling is essential
- **Major issue**: BBV sampling (code-only) can fail for applications with array-indirect memory accesses (a[b[i]])



CPU2017 benchmark	96 cores	128 cores	192 cores
500.perlbench_r	0.99	0.98	0.98
502.gcc_r	1.06	1.05	1.05
505.mcf_r	0.88	0.90	1.03
520.omnetpp_r	1.04	1.06	1.01
523.xalancbmk_r	0.84	0.82	0.80
525.x264_r	0.99	0.99	0.99
531.deepsjeng_r	1.06	1.06	1.08
541.leela_r	0.99	0.98	0.97
548.exchange2_r	1.02	1.02	1.02
557.xz_r	0.91	0.92	0.93

TABLE I
BASELINE SPECRATE CORRELATION FOR AMPEREONE SOCS.

- **523.xalancbmk_r outlier**: 20% underestimation on 192-core SoC
  - Correlation drops from 0.84 -> 0.82 -> 0.80 as core count increases

Our Solution & Key Results

- Memory Access Vectors (MAV) Approach
  - Memory access frequency to different memory regions
  - Microarchitecture-independent: Tracks functional memory access patterns

- Key contributions
  - Detailed characterization of 523.xalancbmk_r
  - Combining BBV and MAV methodology
  - **523.xalancbmk_r correlation** at 192 cores: **80%** → **98%** improvement

Experimental Setup and Methodology

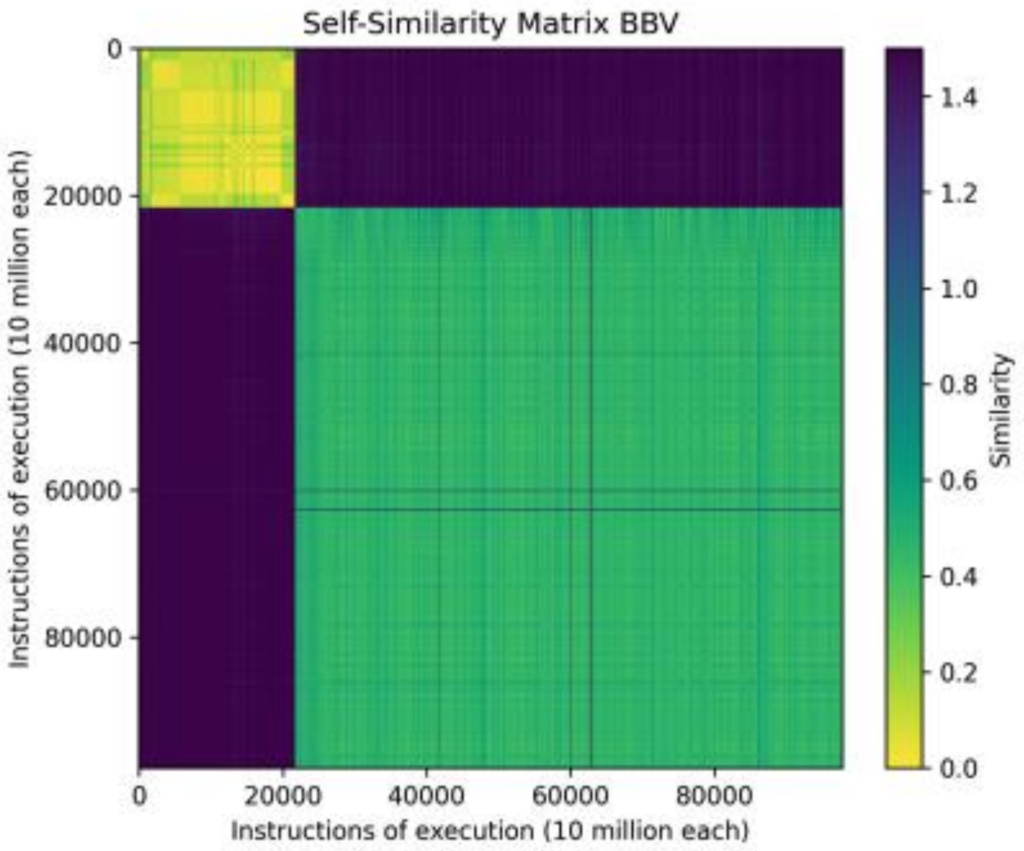
- Setup
  - **Hardware platform**: AmpereOne A192-32X SoC (192 cores)
  - **Performance Model**: In-house simulator targeting AmpereOne architecture
  - **Benchmark:** SPEC CPU2017 integer suite, focus on 523.xalancbmk_r
  - **Trace Collection**: QEMU instrumentation to collect BBV + MAV data

- Correlation methodology
  - **Validation**: Compare performance model scores against hardware measurements
  - **Target**: Correlation as close to 1.00 as possible

Why BBVs Fall Short

- Traditional SimPoint Limitations
  - **Basic Block Vectors**: Count occurrences of code blocks in instruction windows
  - **Assumption**: Code signatures correlate with IPC
  - **Reality**: Same code can exhibit different microarchitectural phases

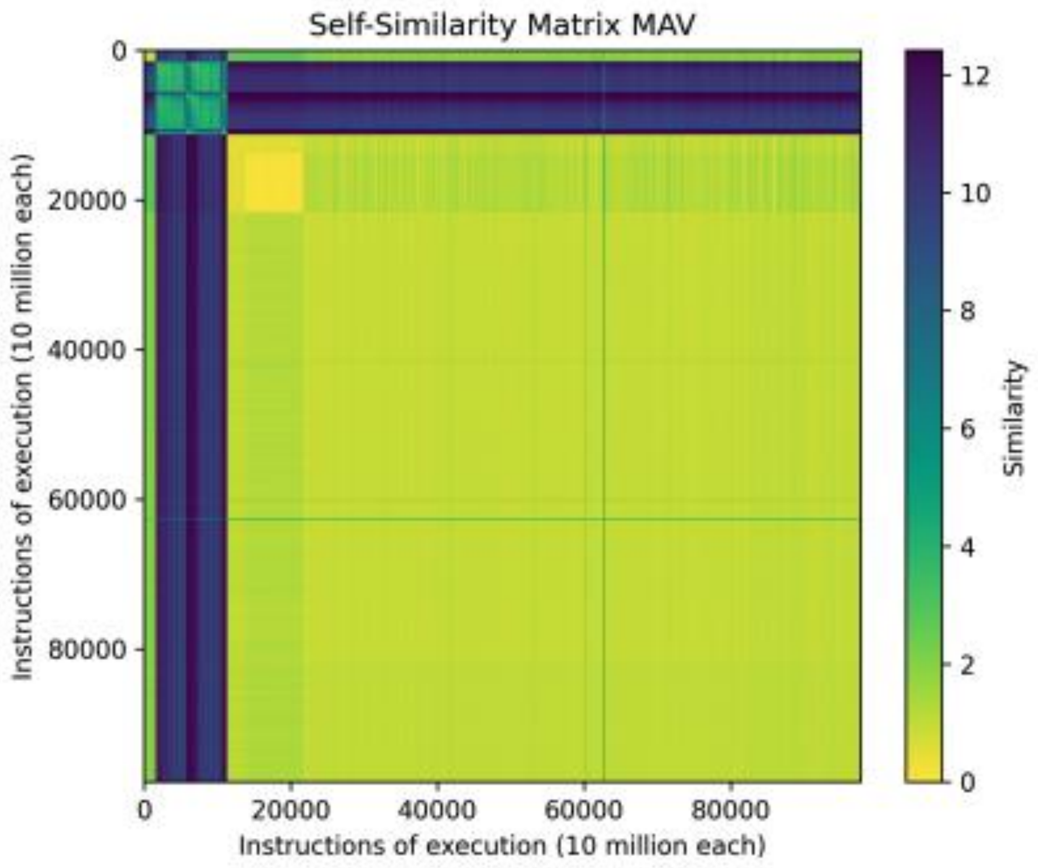
- First ~200B inst: Xerces-C++ 2.7 parser
- Next ~700B inst: Xalan-C++ 1.1 transformer



Self-similarity plot of 523.xalancbmk_r using BBV

Memory Access Vectors (MAV) Methodology

- Core Concept & Design
  - **Microarchitecture independent** tracking of functional memory access patterns
  - **4096-byte granularity** (memory page aligned) in physical address space
  - Records all read/write operations per 10M instruction window
- The 523.xalancbmk_r Case Study
  - Complex data movement patterns
  - Memory access characteristics significantly influence IPC
  - Working set size and access distribution create phase variations



Self-similarity plot of 523.xalancbmk_r using MAV

AMPERE

Combining BBV and MAV

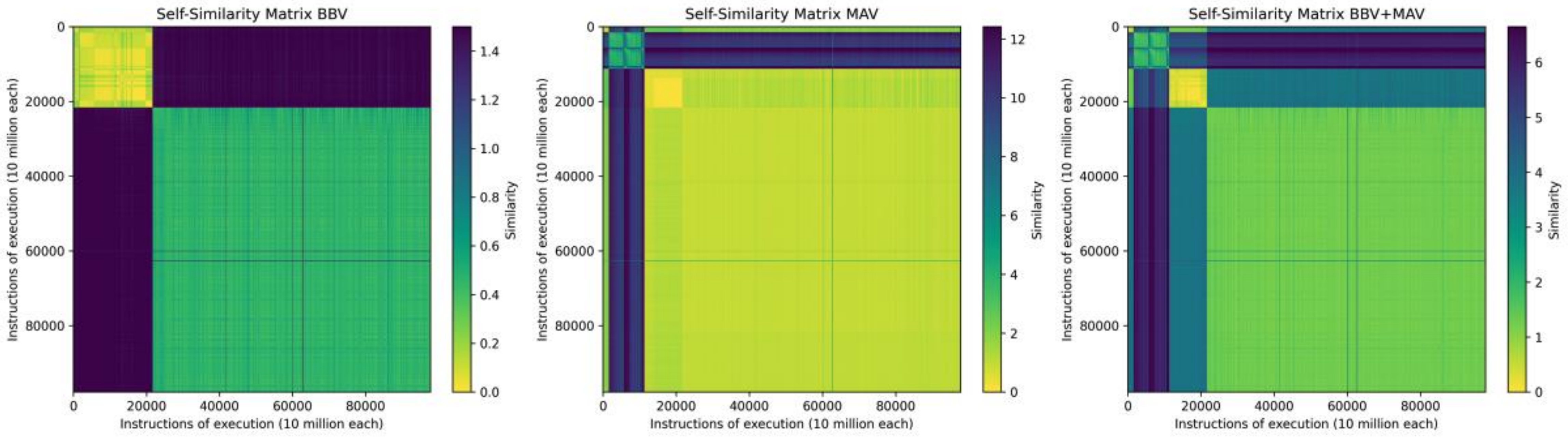


Fig. 1. Self-Similarity plots of 523.xalancbmk_r showing BBV, MAV, and combined BBV+MAV.

BBV alone (left): Shows code **BBV alone** (left): Shows code similarity in first 200B instructions (Xerces parser)

MAV alone (center): Reveals data similarity patterns between 100B-200B instructions

Combined BBV+MAV (right): Identifies multiple phases not visible with either technique alone

Phase Detection Improvement

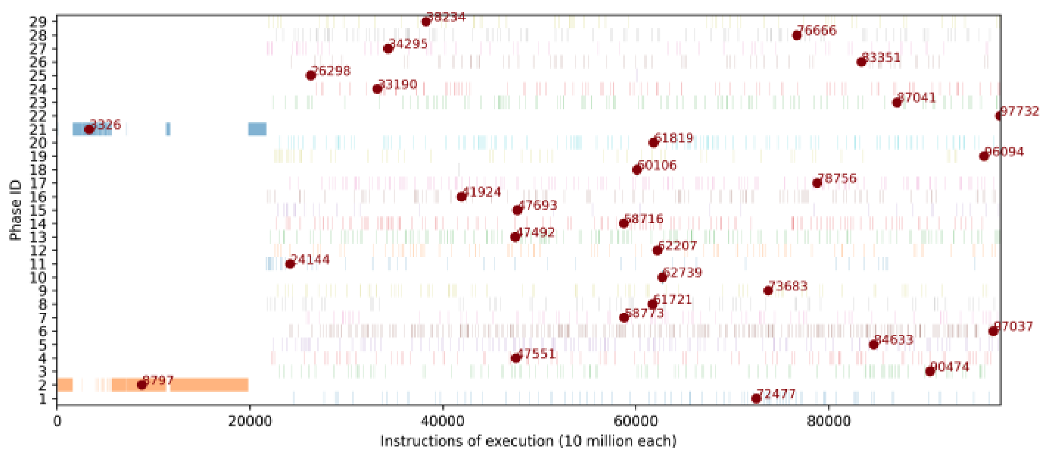


Fig. 2. BBV-only phases and SimPoint selections for 523.xalanc.

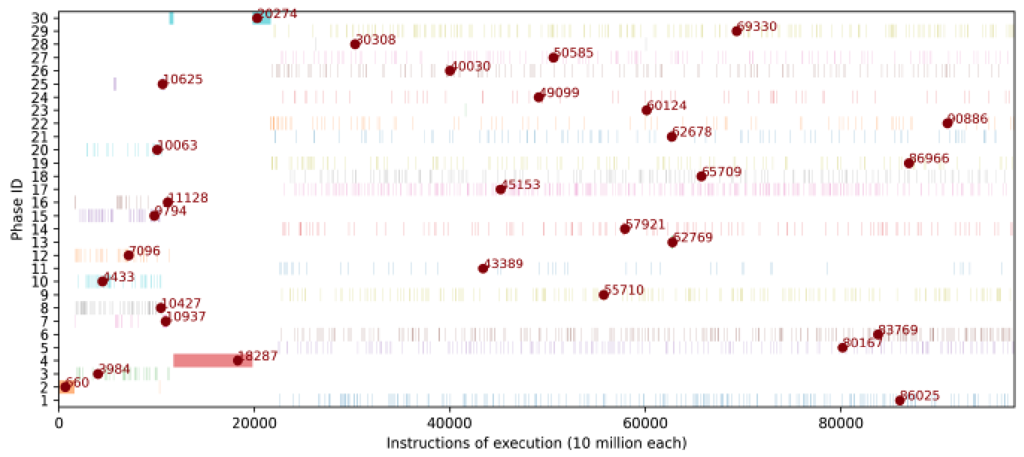


Fig. 3. BBV+MAV phases and SimPoint selections for 523.xalanc.

BBV-Only Limitations (Figure 2)

- Only **2 phases** cover first ~200B instructions (Phase IDs 2, 21)
- Treats Xerces region as homogeneous, inadequate sampling of diverse behaviors

BBV+MAV Enhancement (Figure 3)

- **12 phases** now cover Xerces region (>1/3 of total clusters)
- Clearer separation for k-means clustering, better representation of microarchitectural diversity

Phase Detection Improvement

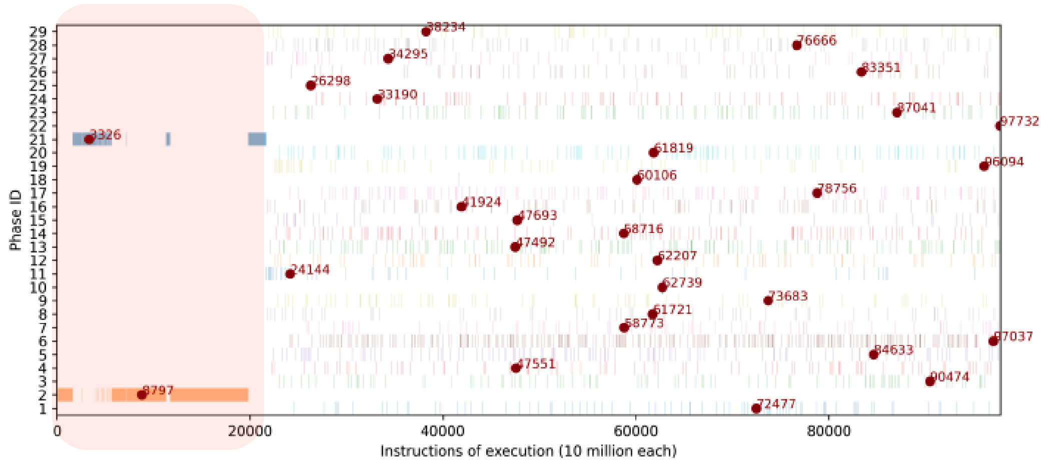


Fig. 2. BBV-only phases and SimPoint selections for 523.xalanc.

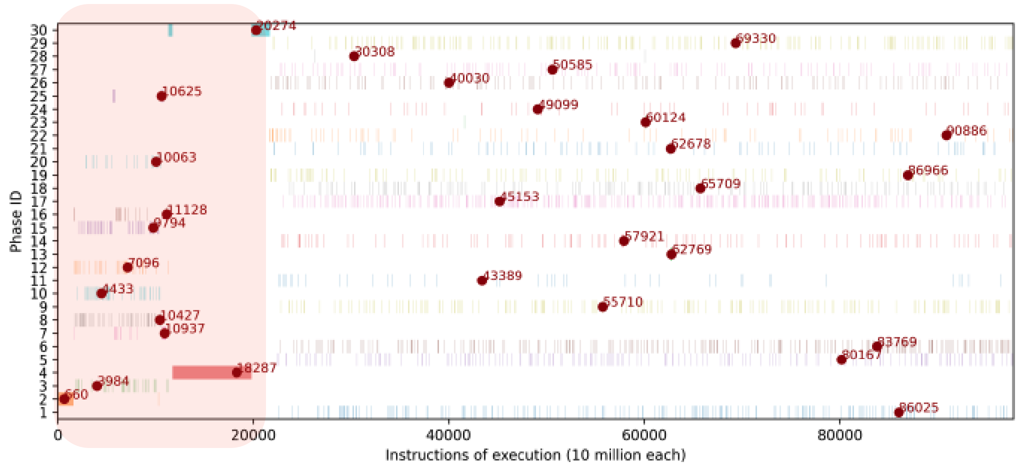


Fig. 3. BBV+MAV phases and SimPoint selections for 523.xalanc.

BBV-Only Limitations (Figure 2)

- Only **2 phases** cover first ~200B instructions (Phase IDs 2, 21)
- Treats Xerces region as homogeneous, inadequate sampling of diverse behaviors

BBV+MAV Enhancement (Figure 3)

- **12 phases** now cover Xerces region (>1/3 of total clusters)
- Clearer separation for k-means clustering, better representation of microarchitectural diversity

Phase Detection Improvement

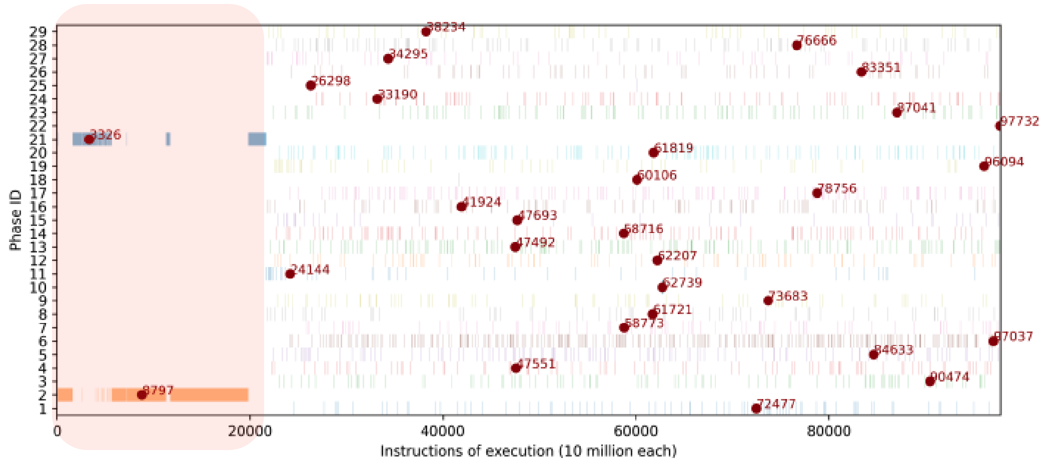


Fig. 2. BBV-only phases and SimPoint selections for 523.xalanc.

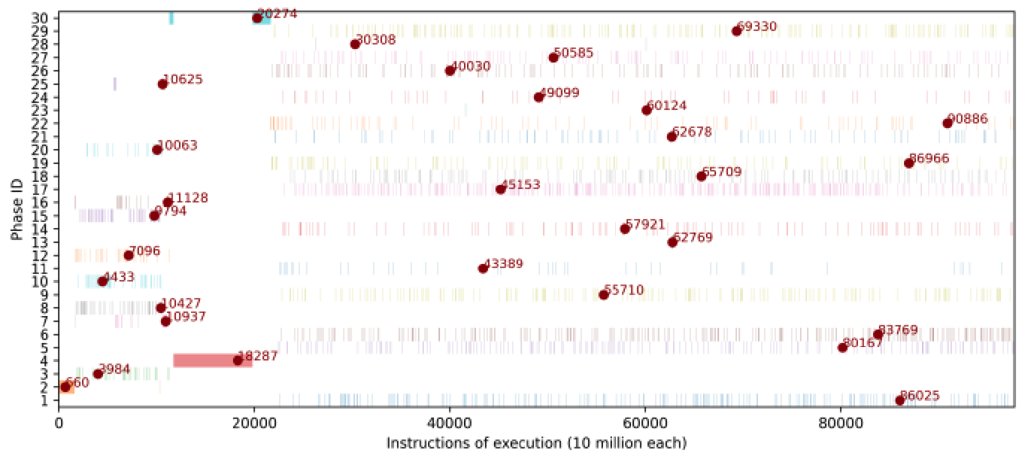


Fig. 3. BBV+MAV phases and SimPoint selections for 523.xalanc.

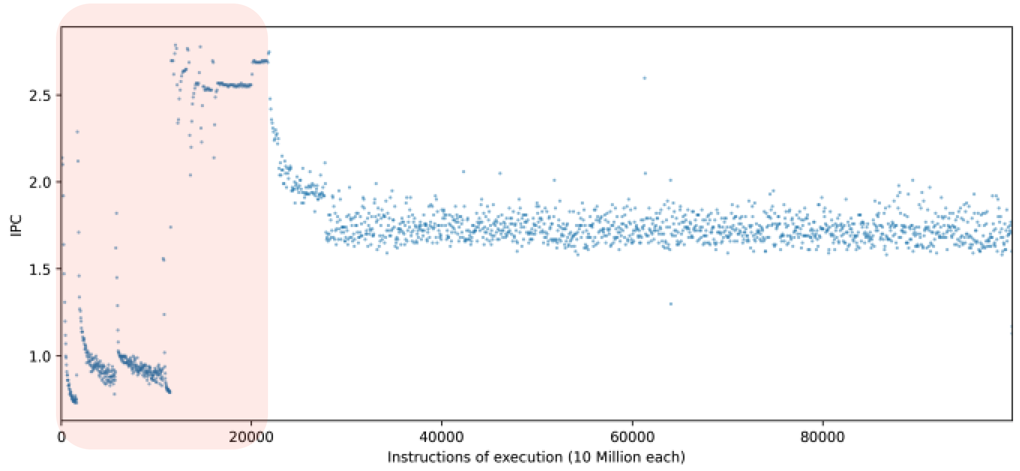


Fig. 4. IPC plot of 523.xalanc on AmpereOne silicon.

Comparison Against Silicon IPC

- Phase 2 (BBV-only) covered regions with both very low and very high IPC
- BBV+MAV Phases 4 and 30 now represent highest IPC regions accurately

Phase Detection Improvement

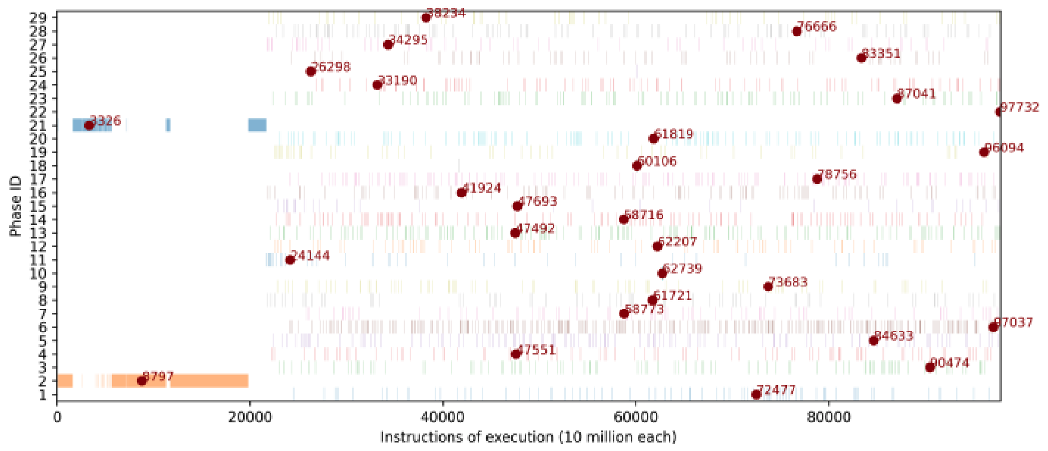


Fig. 2. BBV-only phases and SimPoint selections for 523.xalanc.

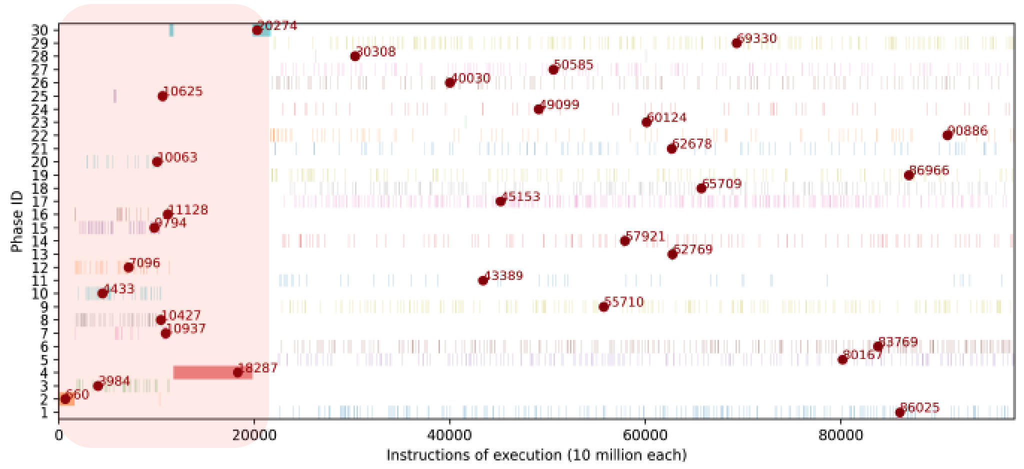


Fig. 3. BBV+MAV phases and SimPoint selections for 523.xalanc.

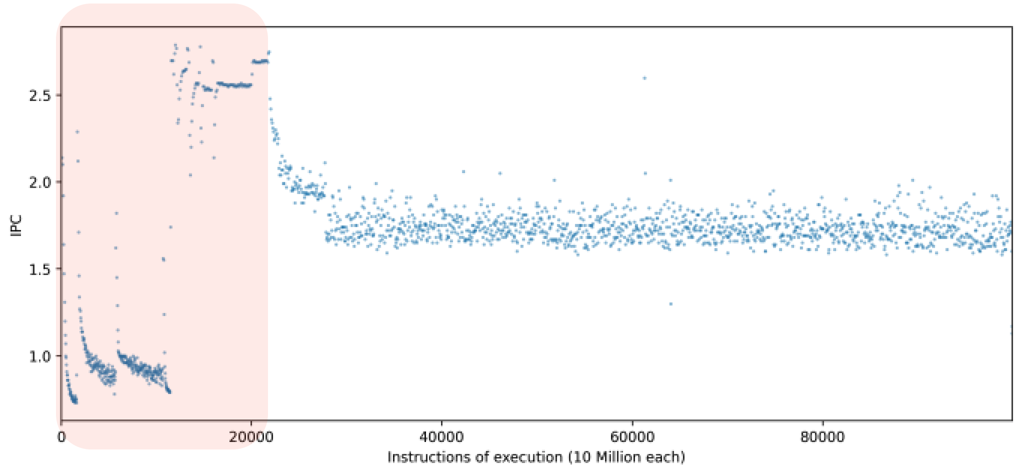
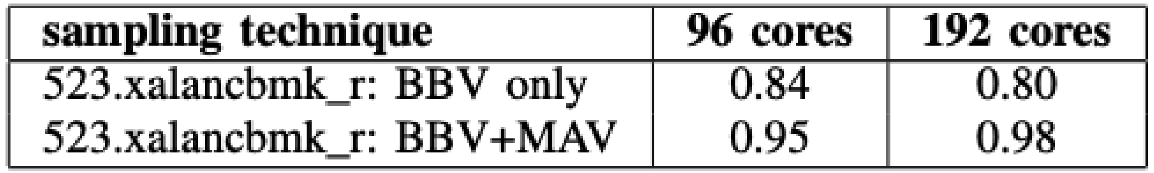


Fig. 4. IPC plot of 523.xalanc on AmpereOne silicon.

Comparison Against Silicon IPC

- Phase 2 (BBV-only) covered regions with both very low and very high IPC
- BBV+MAV Phases 4 and 30 now represent highest IPC regions accurately

Results



sampling technique	96 cores	192 cores
523.xalancbmk_r: BBV only	0.84	0.80
523.xalancbmk_r: BBV+MAV	0.95	0.98

TABLE II
CORRELATION OF BOTH SAMPLING TECHNIQUES ON AMPEREONE SOCS.

- **18%** and **11%** correlation improvement at **192** and **96** cores respectively

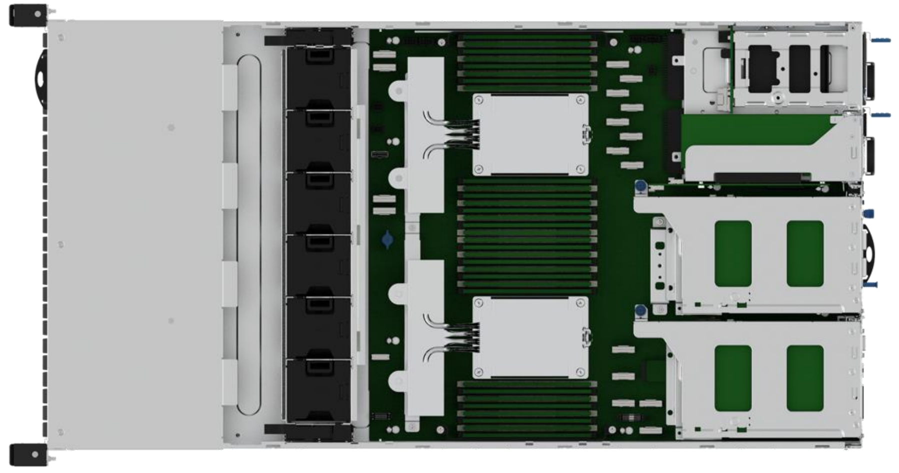
Results and Contributions

- **Workload Characterization:** Identified why 523.xalancbmk_r misses target - parser accesses diverse data despite recurring code, creating microarchitectural phases BBV cannot detect
- **Combining MAV with BBV:** Memory access vectors with performance-focused transformation and automatic adaptive weighting - no manual tuning
- **Correlation Improvement:** 18% and 11% correlation improvement at 192 and 96 cores respectively

Thanks!

Server Seeding program

Ampere donates high core count ARM servers to academic research labs. If you are interested and have rack space available, please email: **mahesh@amperecomputing.com**



- 2P Mt Collins L10 System
- 1x Mt Collins 2U24 NVMe MP
- 2x Q80-30 SoCs (80 core, 3.0Ghz)
- 2x 16GB DIMMs
- 1x 960GB PCIe M.2 SSD
- Onboard NIC
- No warranty or technical support
- Terms and conditions
  - Receiving party must sign Ampere Donor Form
  - Promotion valid while supplies last

MAV Integration Pipeline

- Vector Transformation Process
  - **Inverse Frequency Computation**: 1/access_frequency for each memory region
    - Emphasizes infrequently accessed regions (likely cache misses/page faults)
    - De-emphasizes frequently accessed regions (likely cached)
  - **Performance-Focused Sorting**: Sort by inverse frequencies (descending order)
    - Prioritizes performance-critical memory accesses
    - Memory address labels discarded - focus on access pattern, not location
  - **Result**: Signature captures actual performance impact rather than raw access counts
- **Normalization**: Matrix-wide normalization (vs. individual vector normalization)
  - MAV vectors normalized to have an *average* magnitude of 1
  - Preserves relative memory intensity across instruction windows
- **Temporal Locality**: 0.95 exponential decay over 10 instruction windows
  - Captures longer-term memory reuse patterns
  - Accounts for extensive cache hierarchies in server-class CPUs
- **Dimension Reduction**: Gaussian Random Projection to 15 dimensions
  - Matches BBV dimensionality for equal weighting
  - Combined into 30-dimensional representation (BBV + MAV)
- **Adaptive Weighting**: MAV contribution scaled by % of memory operations
  - Memory-intensive apps: MAVs significantly influence phase detection
  - Compute-bound apps: BBVs remain primary indicator
  - **Automatic adaptation** - no manual tuning required
- **Final Clustering**: Combined BBV+MAV matrix → SimPoint k-means algorithm

AMPERE