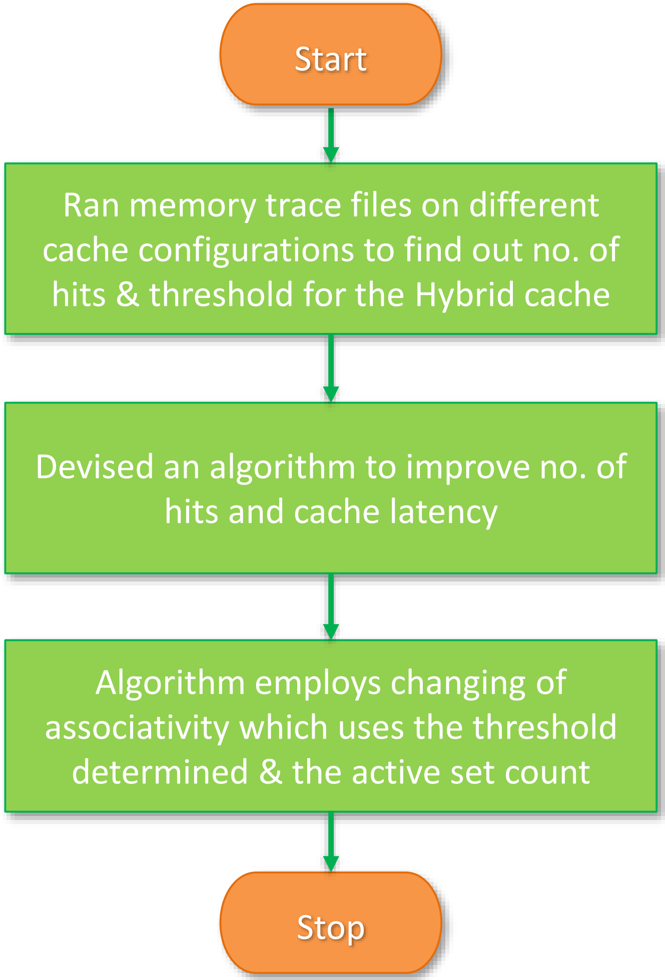
ERIK JONSSON SCHOOL OF ENGINEERING AND COMPUTER SCIENCE
The University of Texas at Dallas

UT DALLAS

HYBRID CACHE

Ashish G Pai	Chris Vinn Mathew	Rakshith Kaushik T R	Sriee Gowthem Raaj A S
(agp150130)	(cxm154130)	(rxt160030)	(sxa156930)

OVERVIEW



Start

Ran memory trace files on different cache configurations to find out no. of hits & threshold for the Hybrid cache

Devised an algorithm to improve no. of hits and cache latency

Algorithm employs changing of associativity which uses the threshold determined & the active set count

Stop

Figure 1: Overview

CACHE CONFIGURATIONS

Parameter	Size
Associativity	1,2 & 4
Block Size (Bytes)	16
Cache Size (Bytes)	32K, 64K, 128K, 256K, 512K, 1M, 2M

- The cache is designed for an uniprocessor system employing write through policy
- A simulated external memory has been used to demonstrate the fetching of data in cases of cache misses. The data that will be fetched will be random data
- The trace files consist of load and store instructions
- Addresses are 24-bits long
- The language in which the simulators are written is Java

STATE DIAGRAM

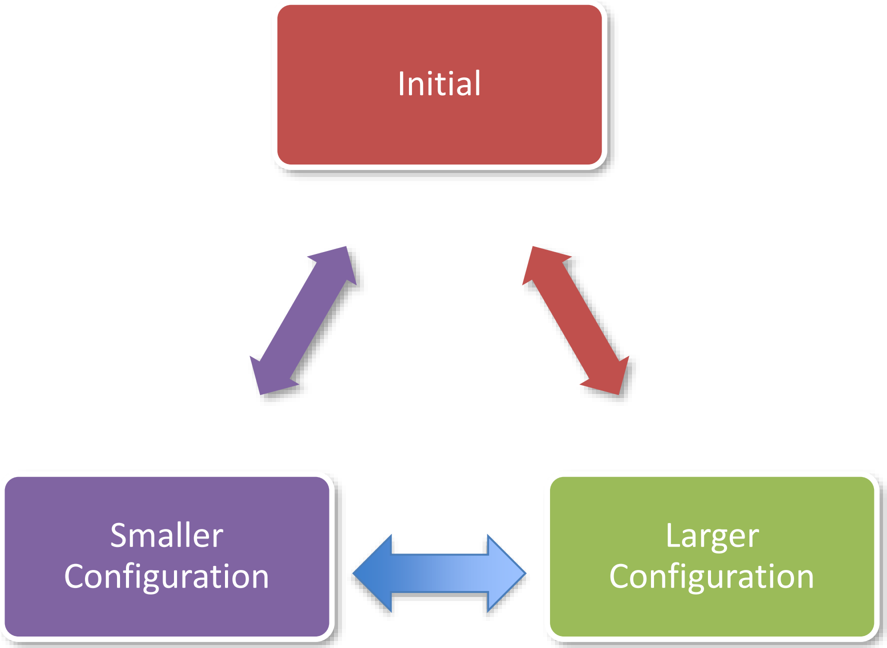


Figure 2: Mode of Operation

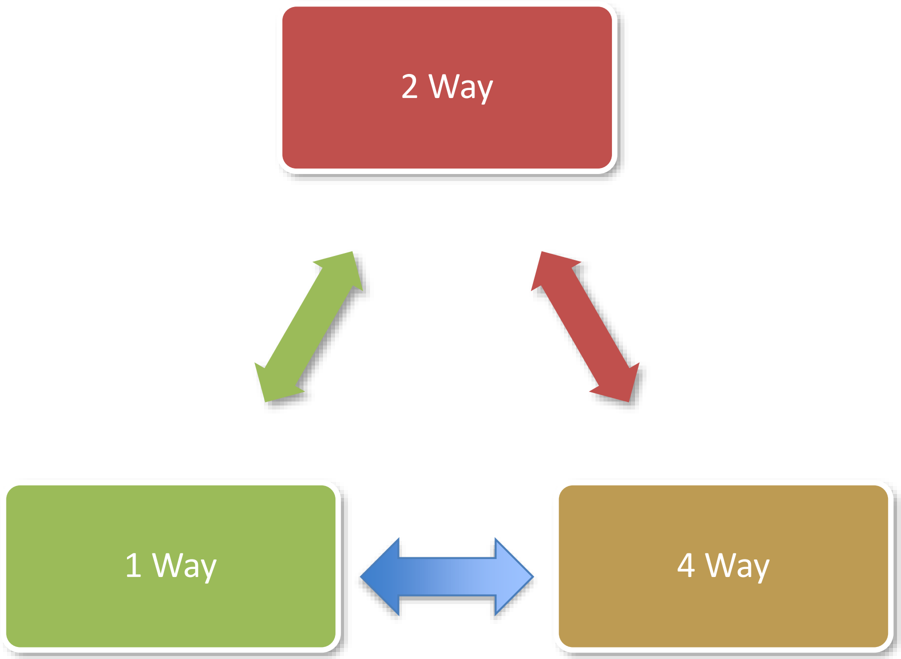


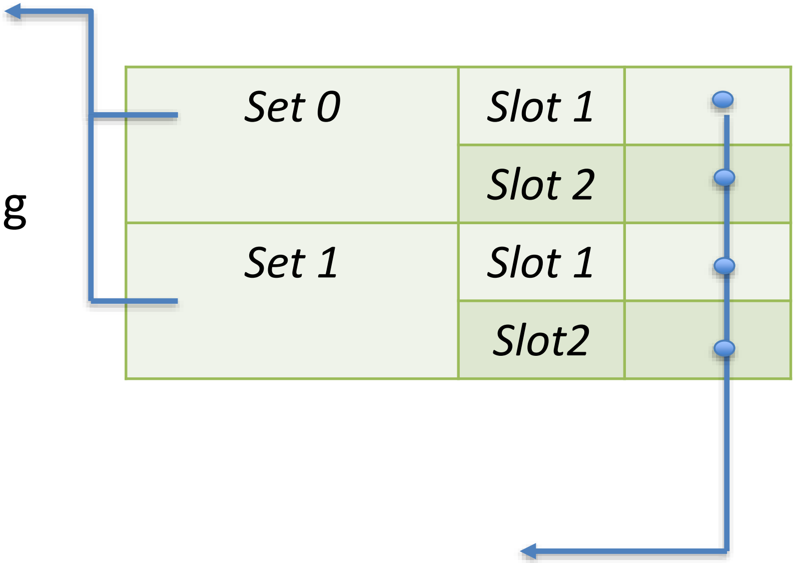
Figure 3: Associativity

HOW DOES HYBRID CACHE WORK?

- Hybrid cache checks for number of misses at periodic intervals
- Check for factors with respect to threshold to reconfigure the cache
- The factors that are considered are Active set count & MRU count
- Cache reconfiguration depends on the decision tree
- The values of the cache are flushed
    - *It initially suffers from Cold cache misses but saved by principle of locality*

Active set count

- Counter for each set
- Incremented every time the set is being accessed
- Used to change the size of the cache

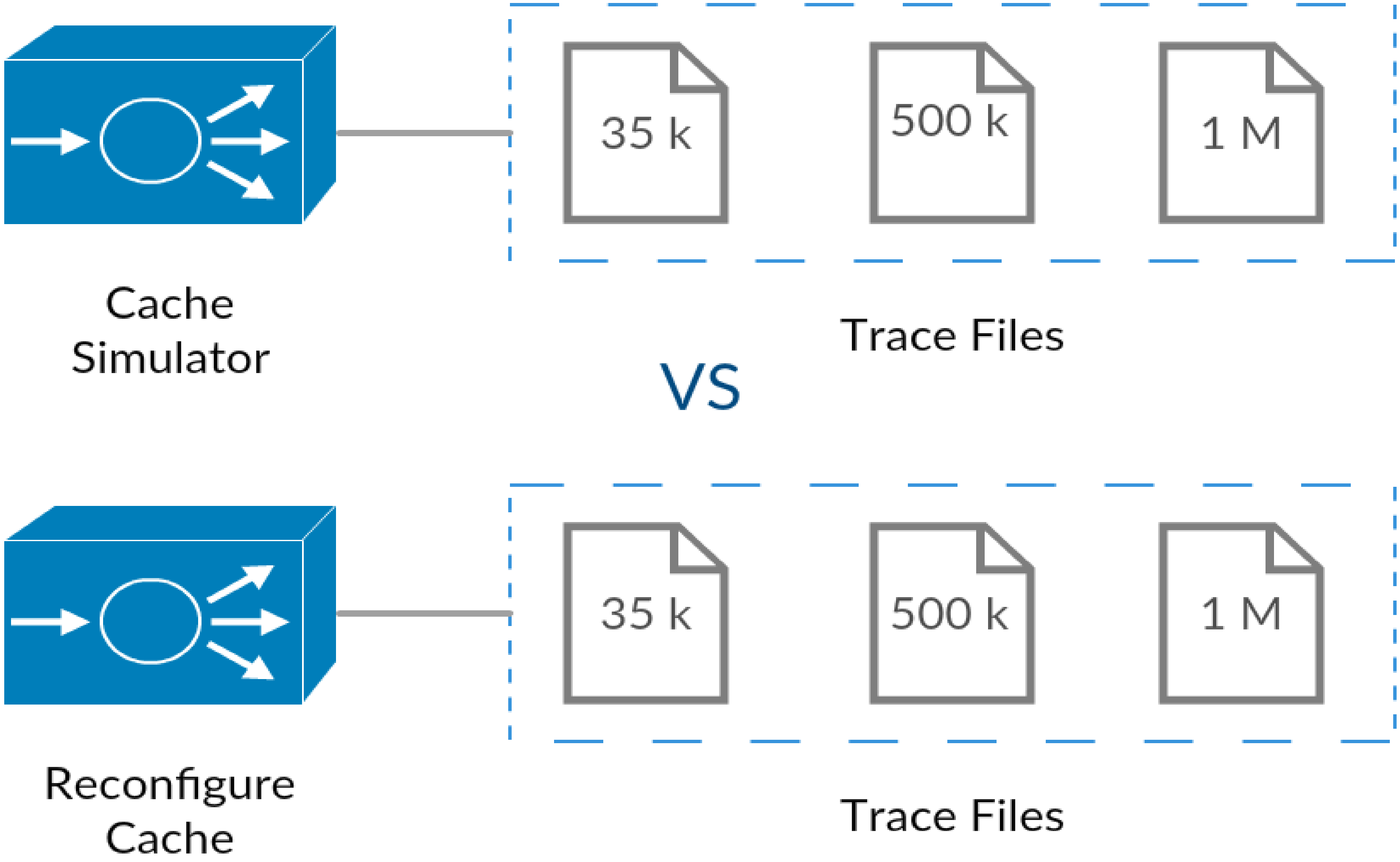


Set 0	*Slot 1*	
	Slot 2	
Set 1	*Slot 1*	
	Slot2	

MRU Count

- The least recently used slot is incremented every time it has been accessed
- Used to change the associativity of the cache

HOW WE SIMULATED?



35 k

500 k

1 M

Cache Simulator

Trace Files

VS

35 k

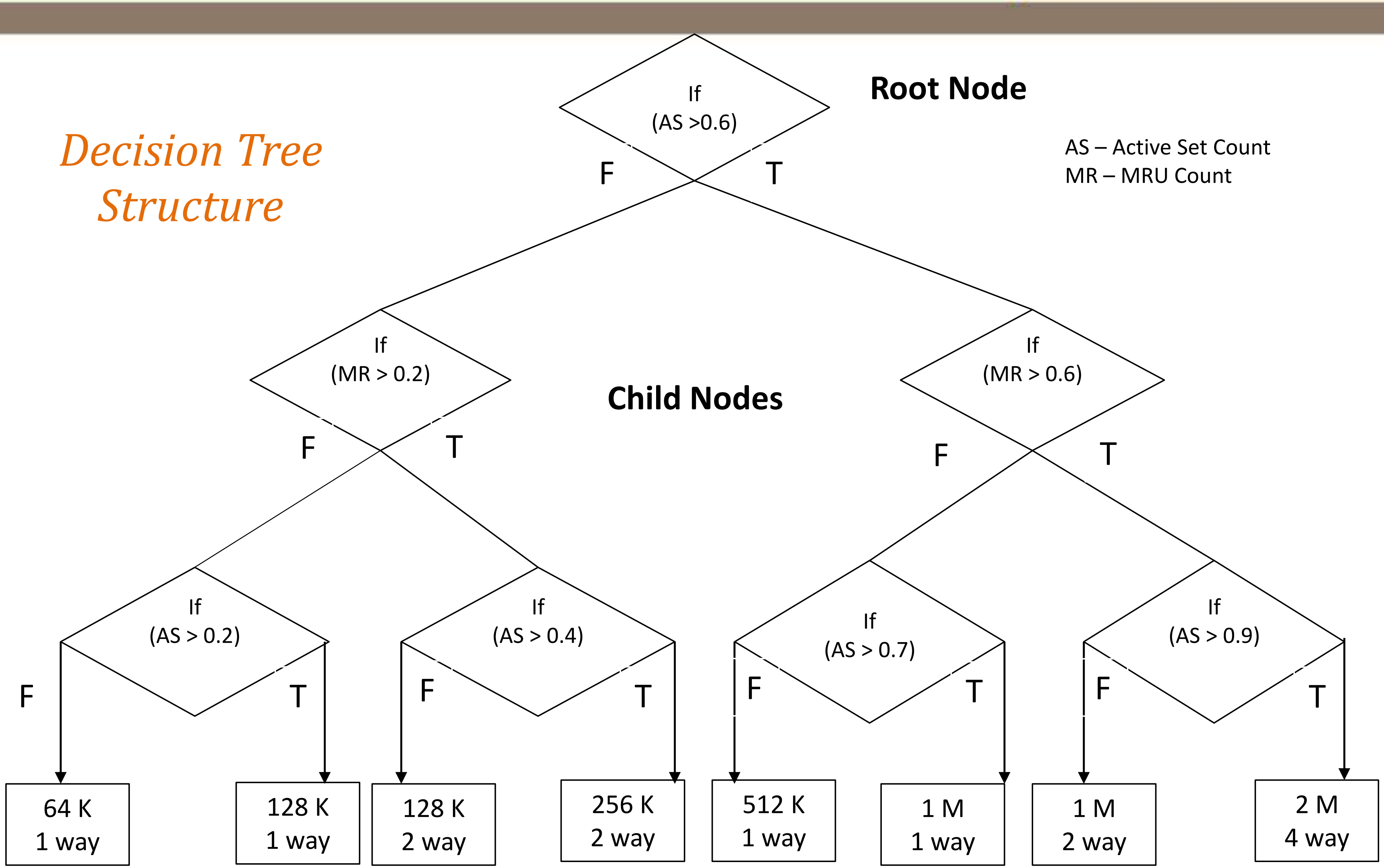
500 k

1 M

Reconfigure Cache

Trace Files

Decision Tree Structure



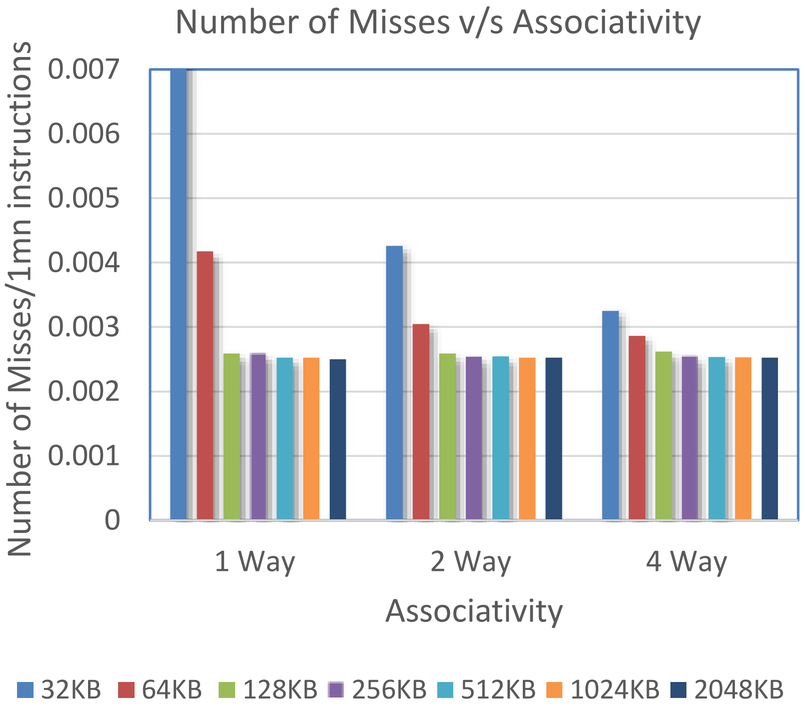
AS – Active Set Count
MR – MRU Count

- *Scenario :*

Associativity	Cache Size	Active Set Count	1765 out 2048 (>0.8)
4	128 K	Total Lines of Instruction	1 Million
		MRU Count	2 out of 4 (=0.5)
		Break Point	10,000

- *Inference*
  - Increased cache size
  - Decrease the associativity
  - Continue execution after breakpoint (10,000)

RESULTS



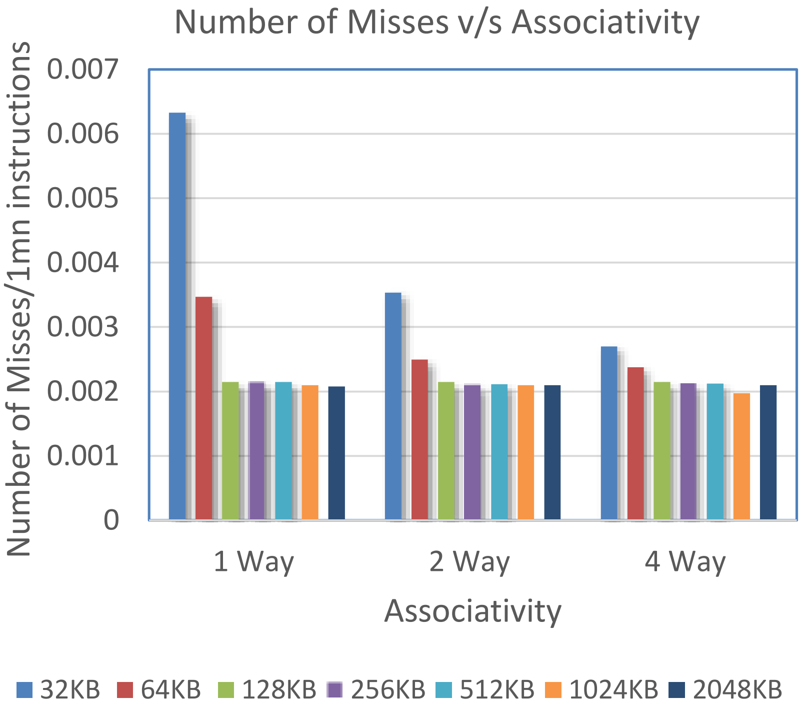


Figure: Results for Normal cache

Figure: Results for Hybrid cache

LESSONS LEARNT

- Software v/s Hardware
- Data sets
- Decision Tree (Generalize)
- Added Functionality

SCOPE FOR FUTURE WORK

- Application can be extended to support other factors that influence cache performance like Energy Delay Product, Power Consumption
- Use other methods to learn the tree
- Validation of data sets
- Extend to other cache variations
  - Instruction & Data cache
  - Replacement Policies
  - Increased Associativity

REFERENCES

1. Karthik Sundararajan, T. Timothy Jones, M. Nigel Topham, P. (2012) The Smart Cache: An Energy-Efficient Cache Architecture Through Dynamic Adaptation. EPSRC.

2. Mattson, R.L., Gecsei, J., Slutz, D.R., Traiger, I.L.: Evaluation techniques for storage hierarchies. IBM Systems Journal. 9(2) (1970)

3. Computer Architecture Research Project. University of Maryland

Thank You !..