eMRC: Efficient Miss Ratio Approximation for Multi-Tier Caching

Zhang Liu
Hee Won Lee
Yu Xiang
Dirk Grunwald
Sangtae Ha



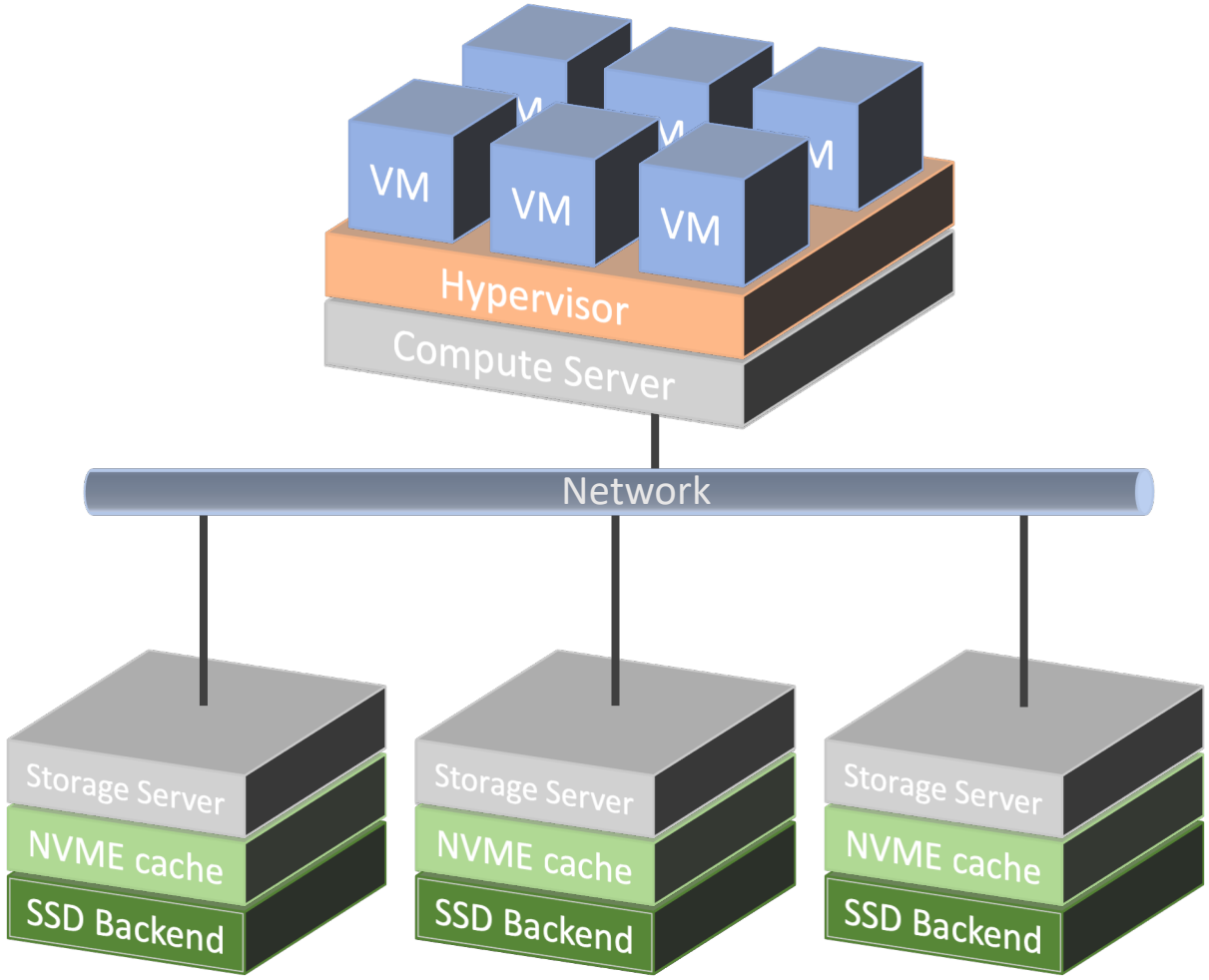
University of Colorado **Boulder**



AT&T Labs Research

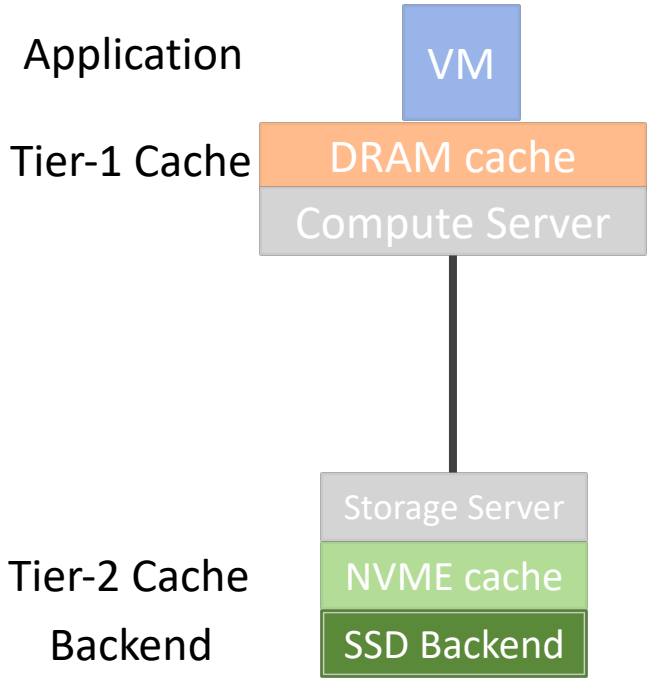
Background

- Today's data centers use multi-tier caching



Background

- Today's data centers use multi-tier caching
- Multiple caching locations
- Workload IO characteristics are important

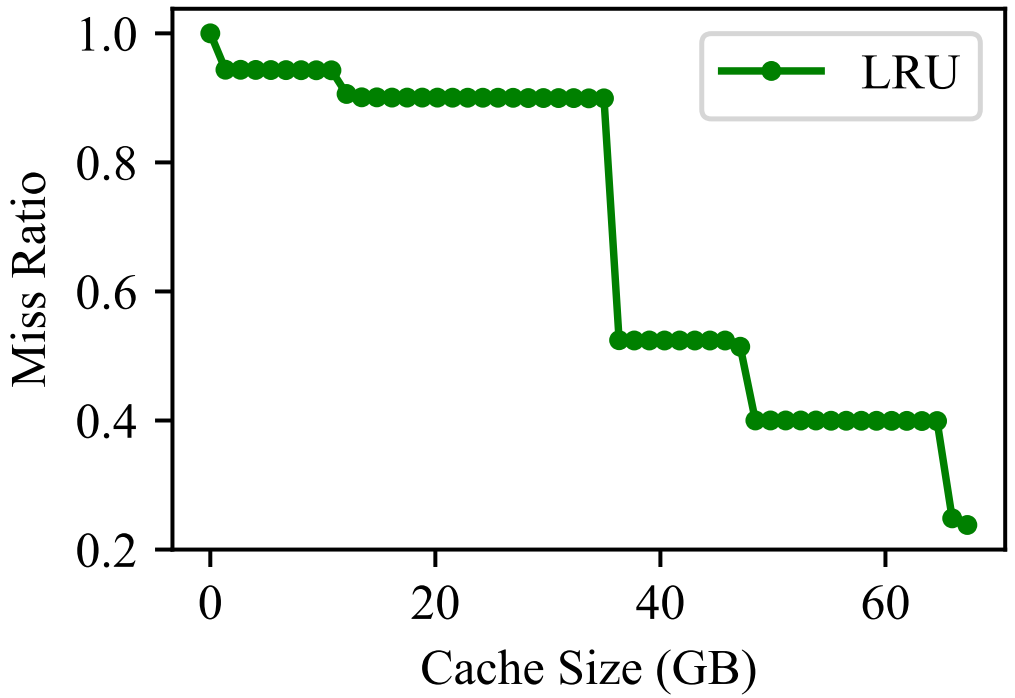


Miss Ratio Curve (MRC)

- Relationship between cache size and resulting miss ratio

- Can we use MRC for multi-tier caching?

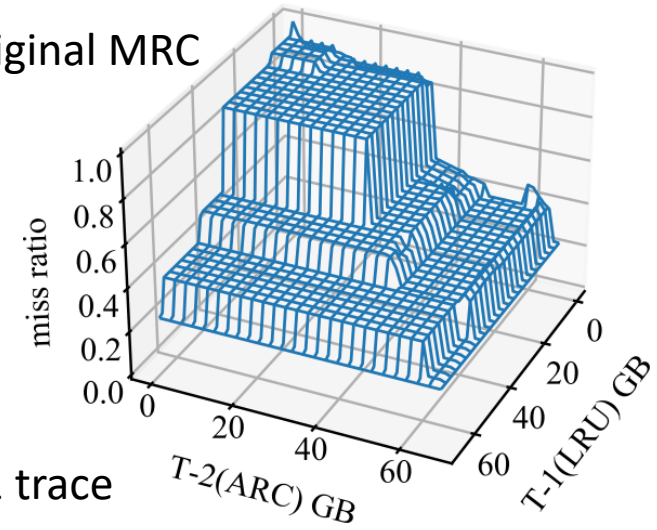
Yes, but with some complications.



Miss Ratio Function for Multi-tier caching

- Slow
- Performance cliffs

Original MRC

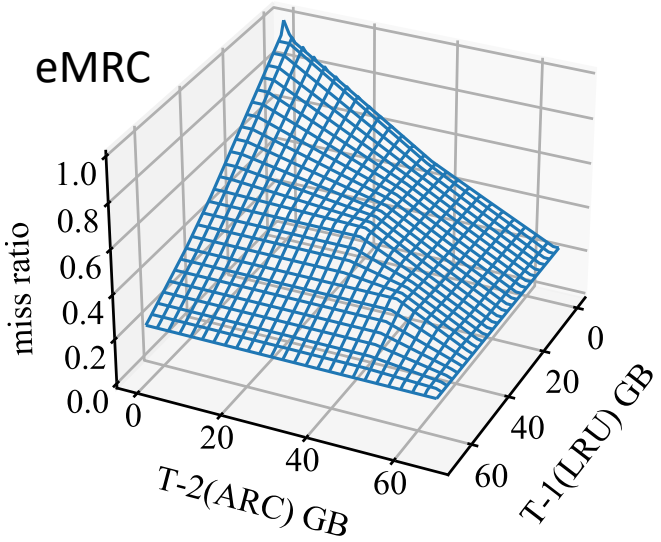


MSR web_2 trace

Computation Time:
1 hour 14 minutes

- Fast
- Better miss ratio

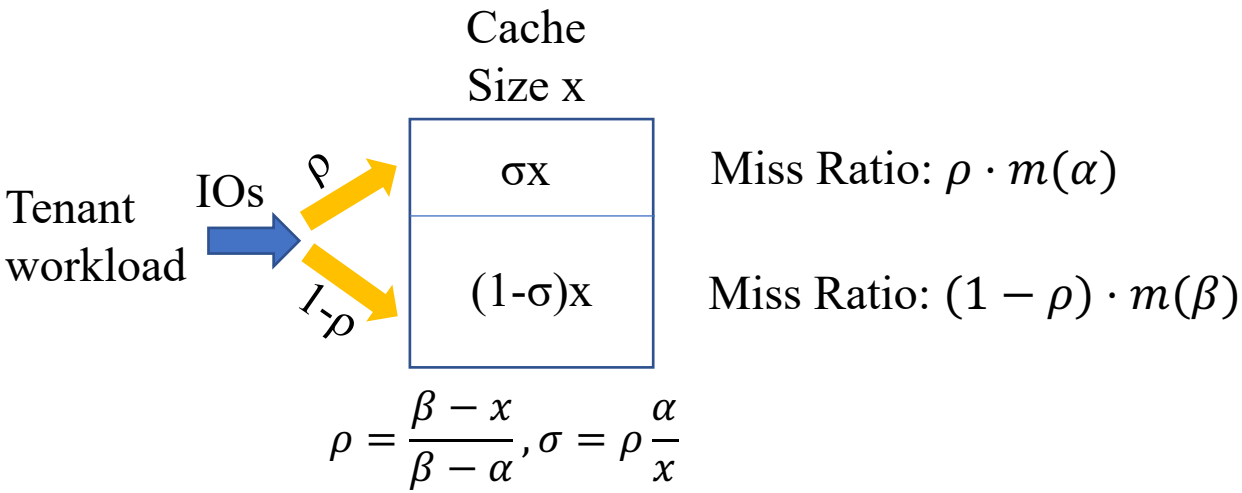
eMRC

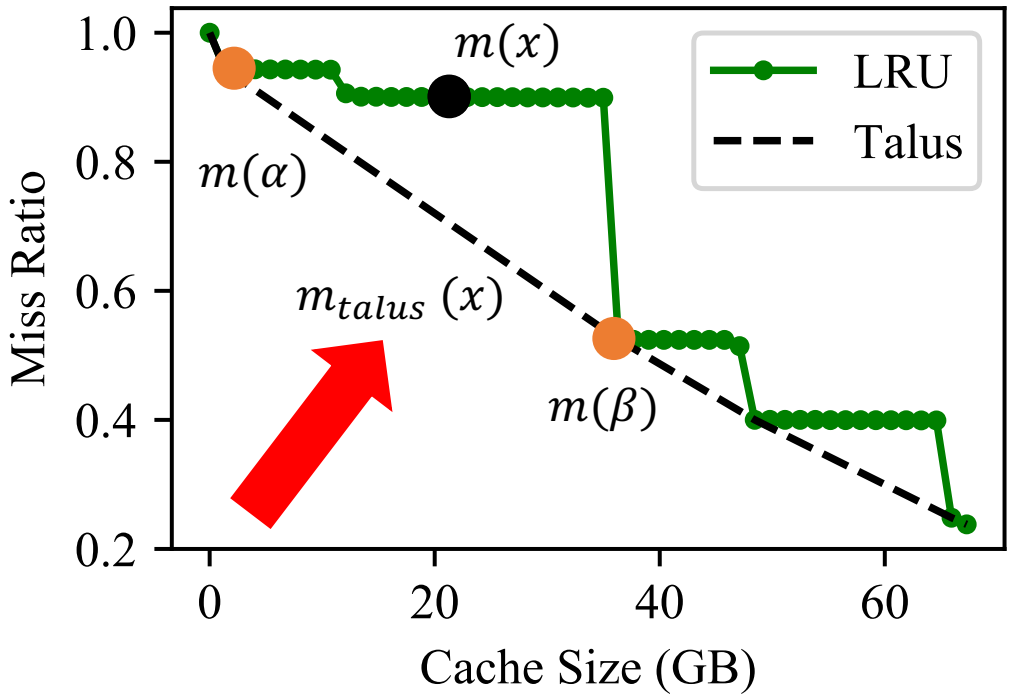


Computation Time:
3 minutes

Talus Cliff Removal (HPCA 2015)

- Talus uses ***spatial sampling*** on the IO and ***shadow partitions*** on the cache to reduce the miss ratio

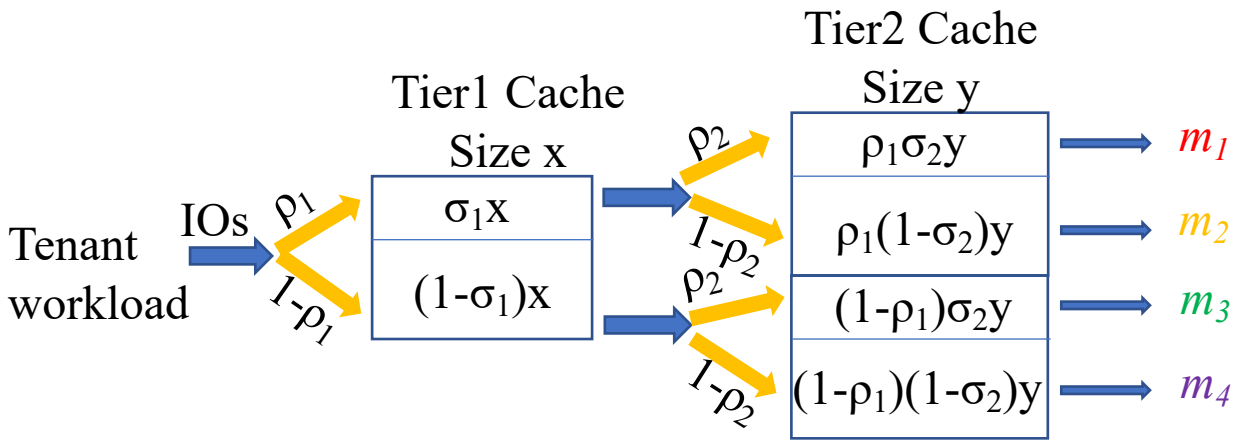


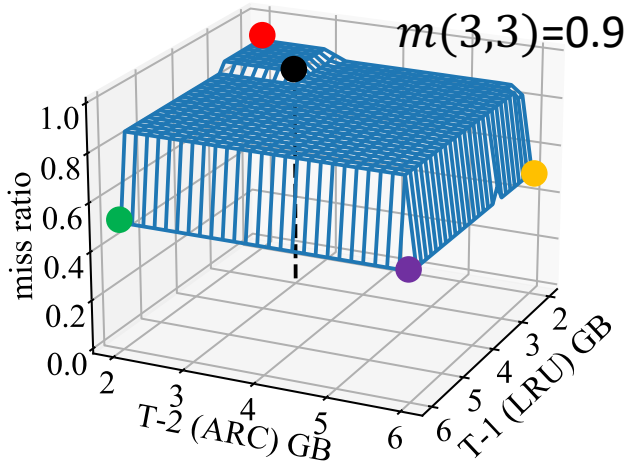


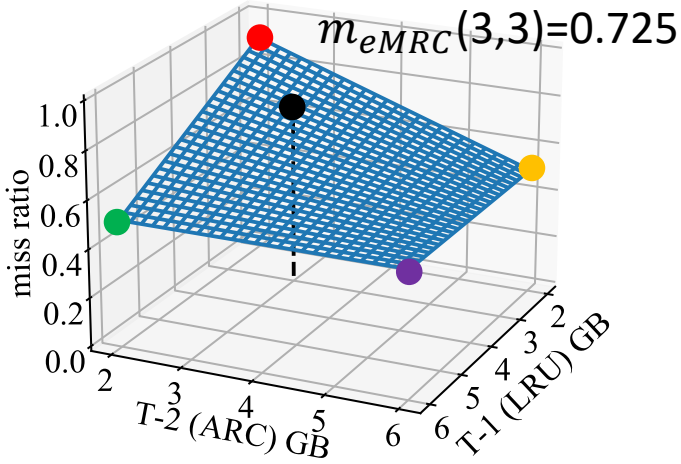
- The miss ratio at each shadow partition is only related to one of the boundary point's miss ratio

eMRC Cliff Removal

- N-tier caching system has 2^N boundary points for a cliff region
- eMRC uses 2^n partitions for n-th tier cache
- The miss ratio at each partition is only related to one of the boundary point's miss ratio





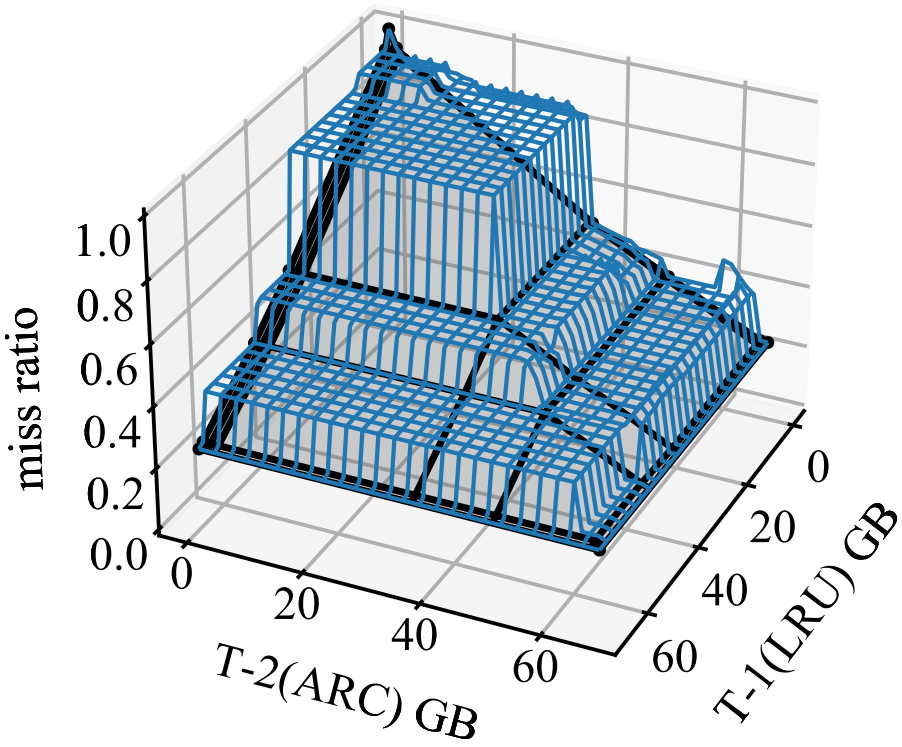


See our paper for proofs and math for N>2

Apply eMRC to the whole space

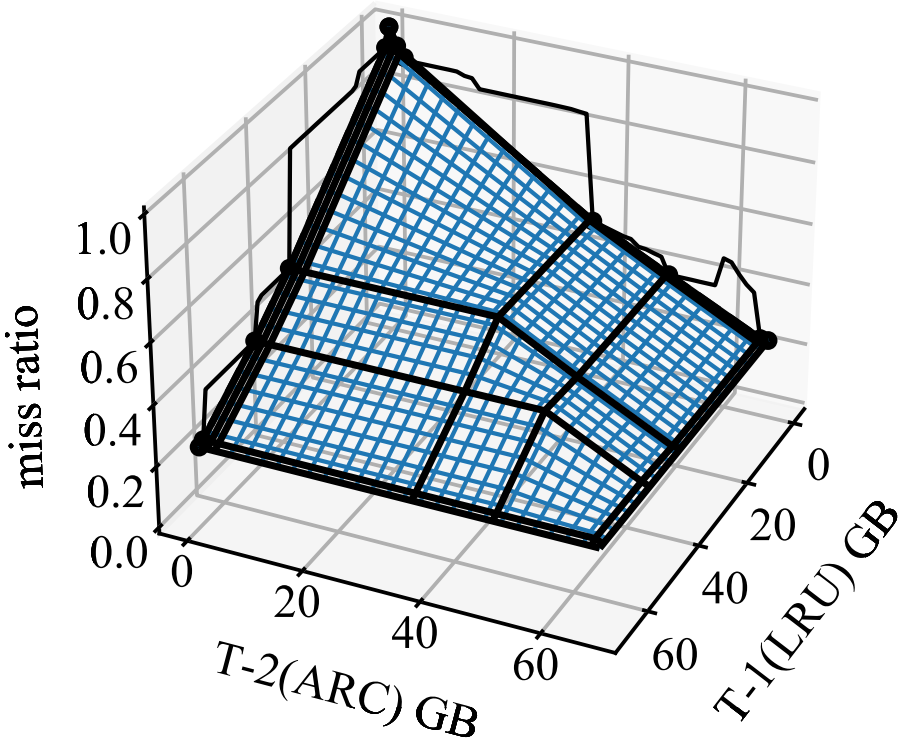
- Need to partition the space into multiple eMRC ***grid*** regions

- Use the ***convex hull*** to generate regions
  - Requires the original MRC as input
  - Time consuming: **1 hour 14 minutes**



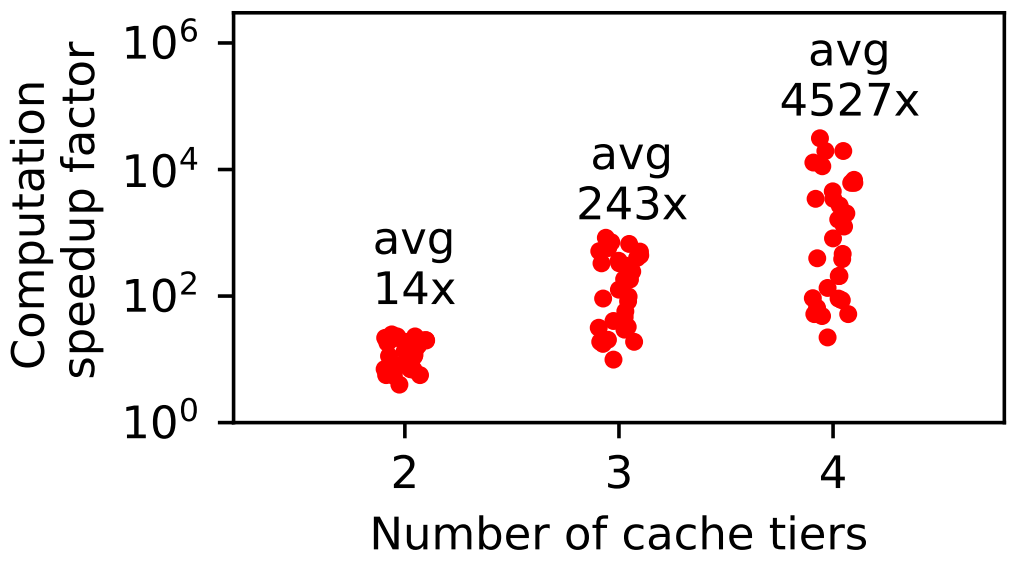
eMRC Convex Hull Approximation

- Our observation using MSR traces:
  - At least 81% vertices of the 3D convex hull are the vertices of the 2D convex hulls on the edges
- Partition the space without knowledge of the whole original MRC

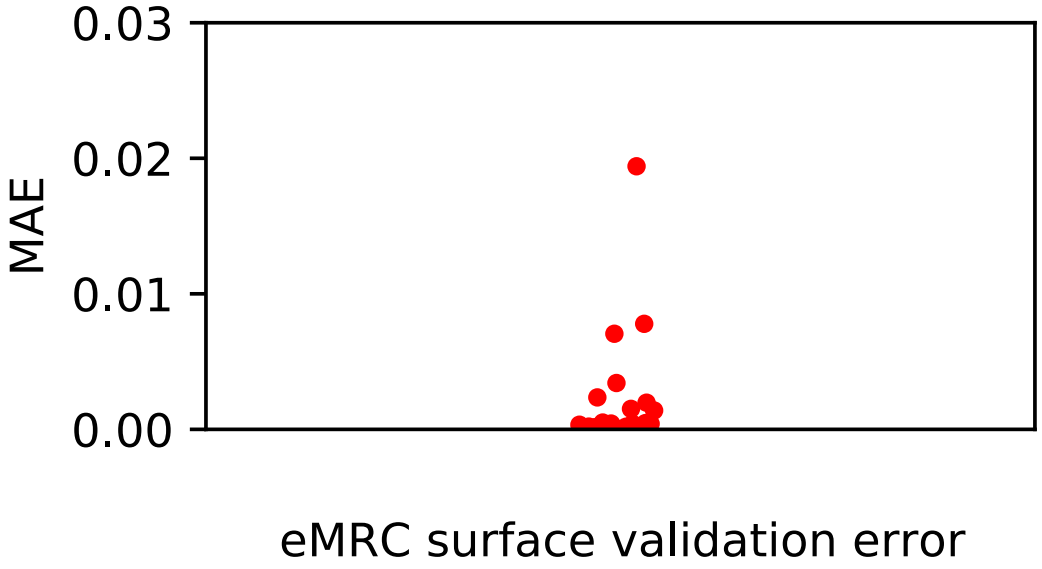


eMRC Performance

- MSR Cambridge IO Traces are used



Comparison of number of datapoints required



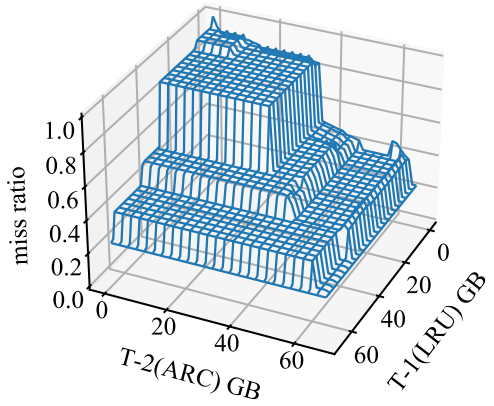
Comparison of eMRC prediction and eMRC validation

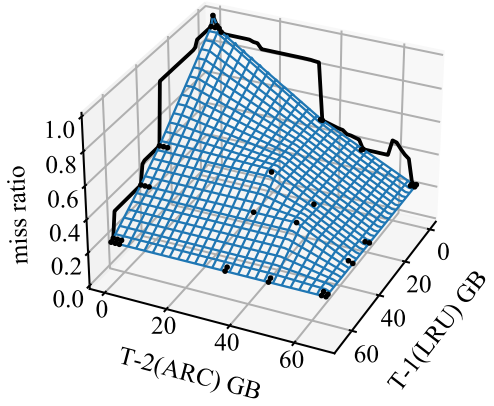
eMRC Performance

Original MRC

eMRC

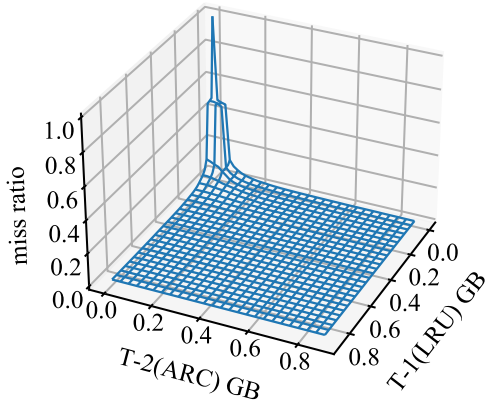
web_2 trace

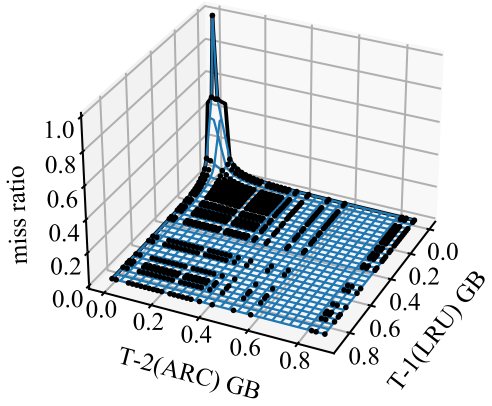




speedup: 20x

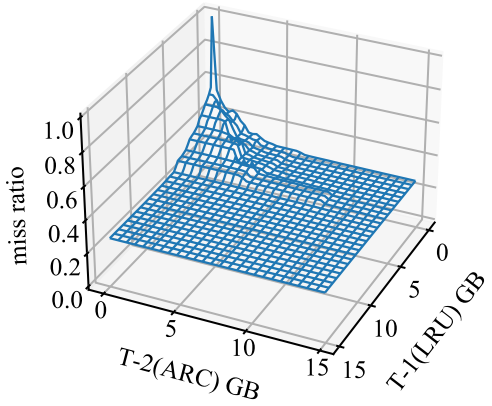
prxy_0 trace

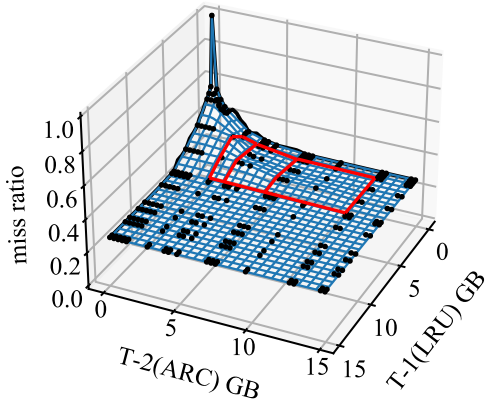




speedup: 4x

prn_0 trace

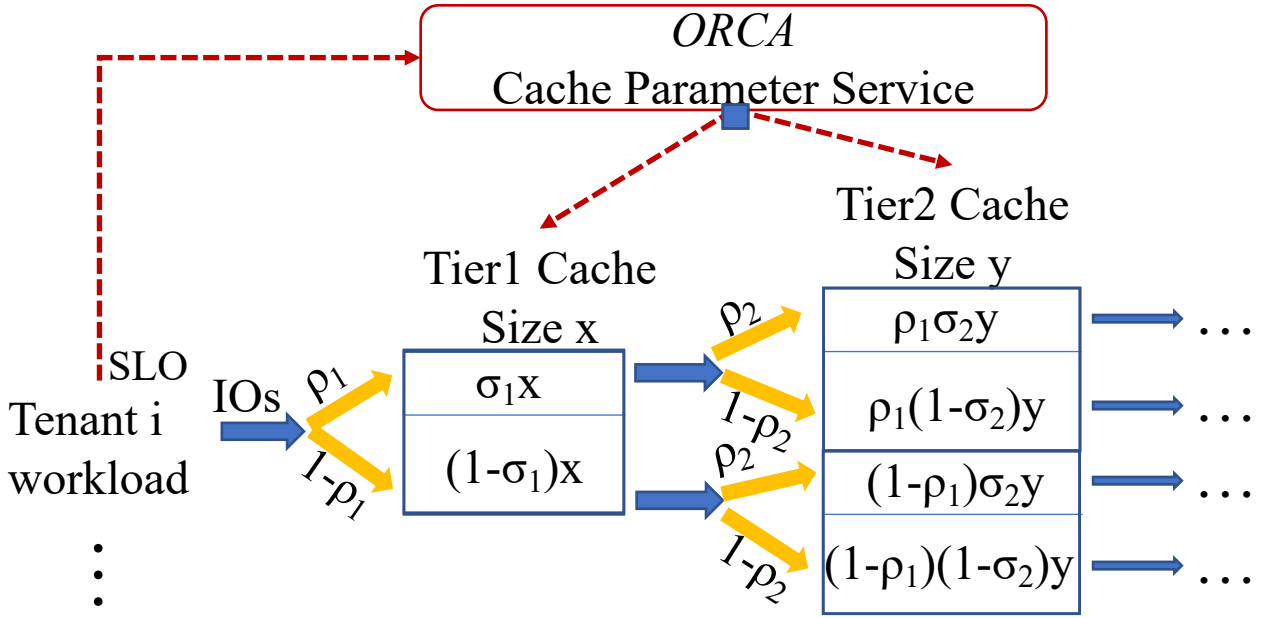




speedup: 8x

ORCA: ORchestration of CAches

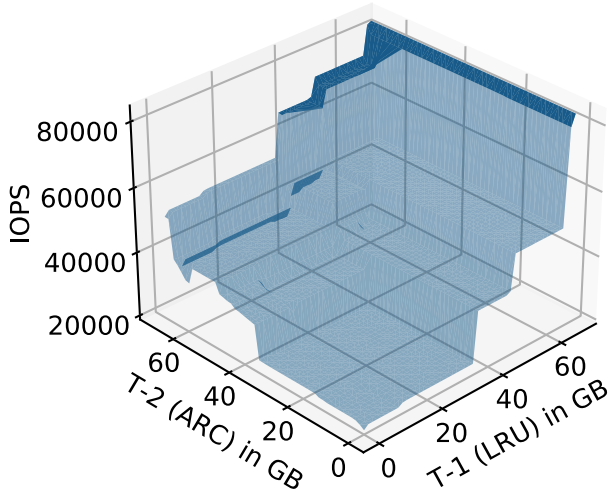
- A multi-tier cache management framework using IOPS as SLO
- Utilizes eMRC to achieve better IOPS performance
- 2-stage algorithm to reduce search space



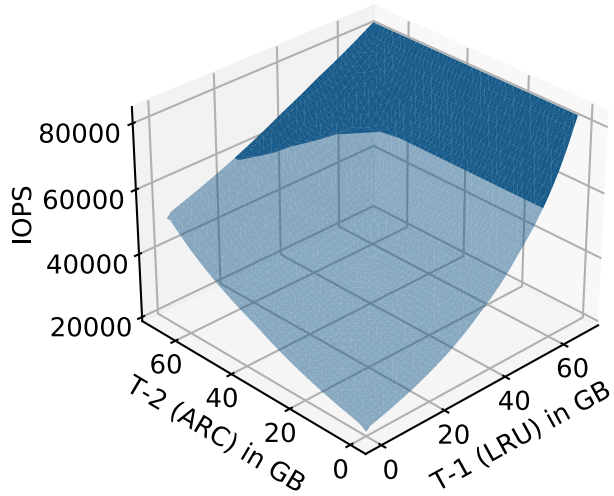
Example

- MSR web_2 workload
- SLO is 60K IOPS
- The Tier-1 cache costs 10x more than the Tier-2 cache
- Find the cache configuration with minimum cost

IOPS without eMRC



IOPS with eMRC



	without eMRC	with eMRC
Tier-1 LRU cache size	38GB	22GB
Tier-2 ARC cache size	65GB	66GB
Time	1hour 14 minutes	3 minutes
Relative cost saving	-	36%

Conclusion

- Multi-dimensional performance cliff removal
  - Works for both stack and non-stack eviction policies
- Efficient MRC generation generation for multi-tier caching
  - Average speedup factor is 14x for 2-tier caching, 4527x for 4-tier caching
- ORCA Cache orchestration framework
  - minimize cache usage for IOPS SLOs

Thanks!

Contact: zhang.liu@colorado.edu