Caching Techniques to Improve Disk IO Performance in Virtualized Systems

Shyamli Rao
153050009

Under the guidance of
Prof. Purushottam Kulkarni

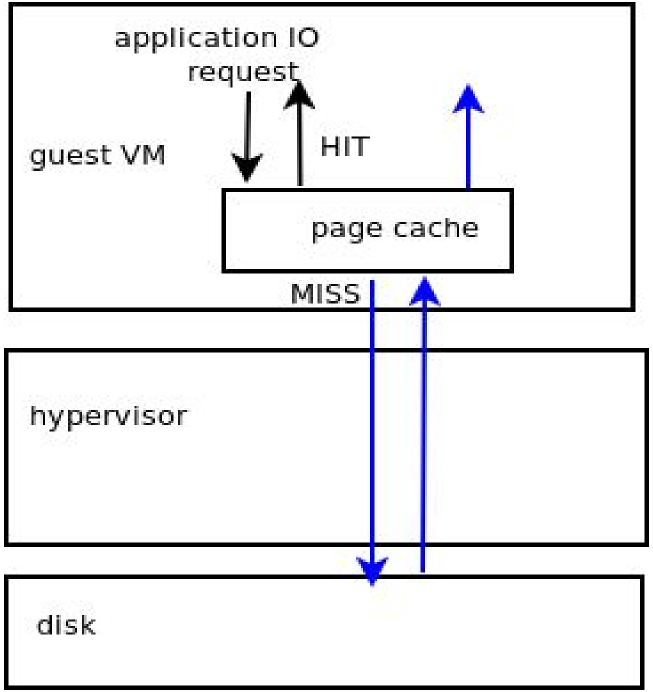
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Problem context

- Disk IO bottleneck

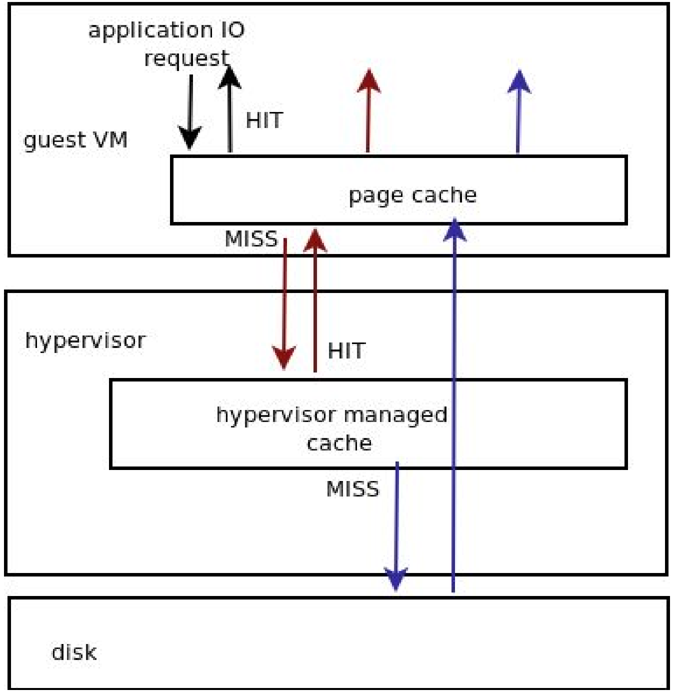
- CPU sits idle

IO flow in virtualized system



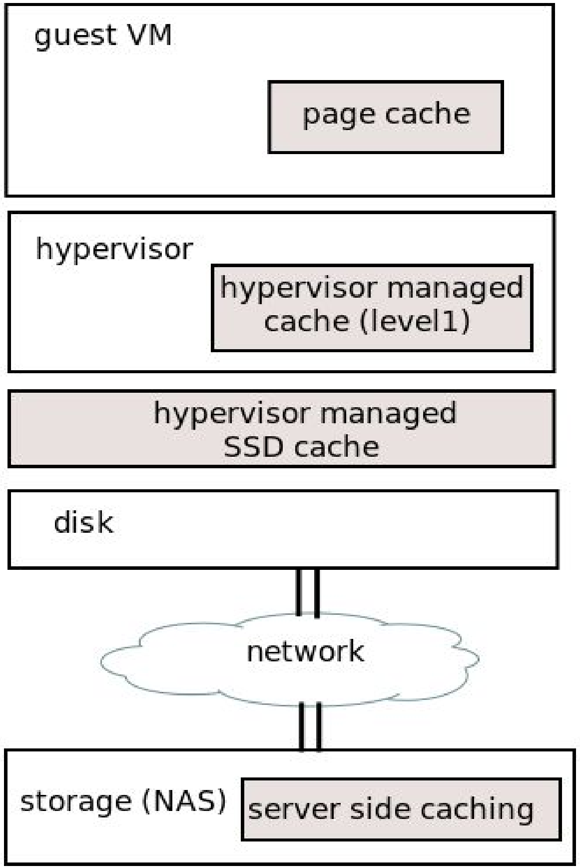
IO flow in hypervisor

Caching - improves disk IO performance



IO request flow with extra cache layer

Cache Configuration



Possible combinations

Cache in guest VM	Hypervisor managed in memory cache	Hypervisor managed SSD cache	Server side cache
✓			
✓	✓		
✓		✓	
✓	✓	✓	
✓		✓	✓

Hypervisor managed caching

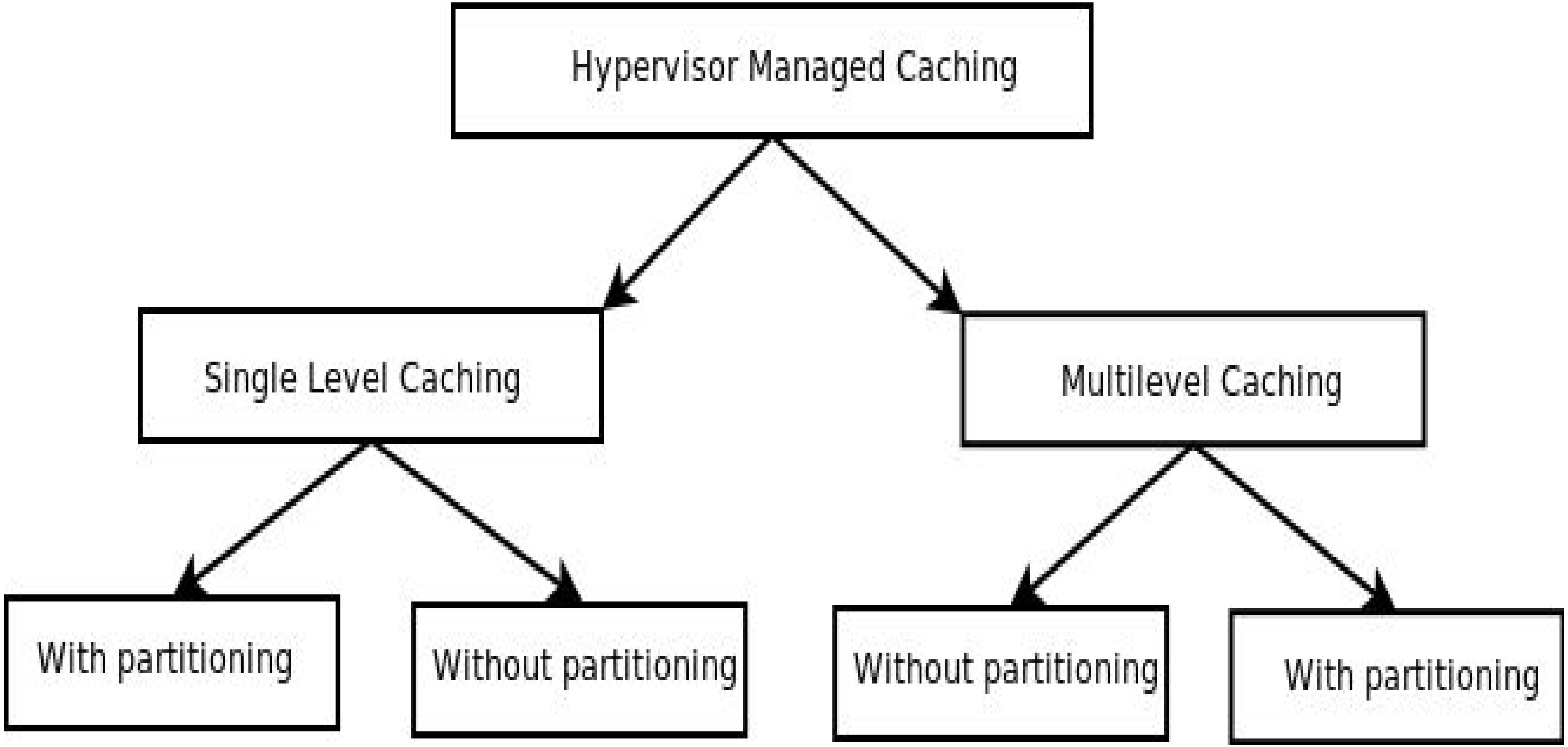
Objectives

- ❑ Improve overall IO performance
- ❑ Fairness
- ❑ Prioritize VMs
- ❑ SLO

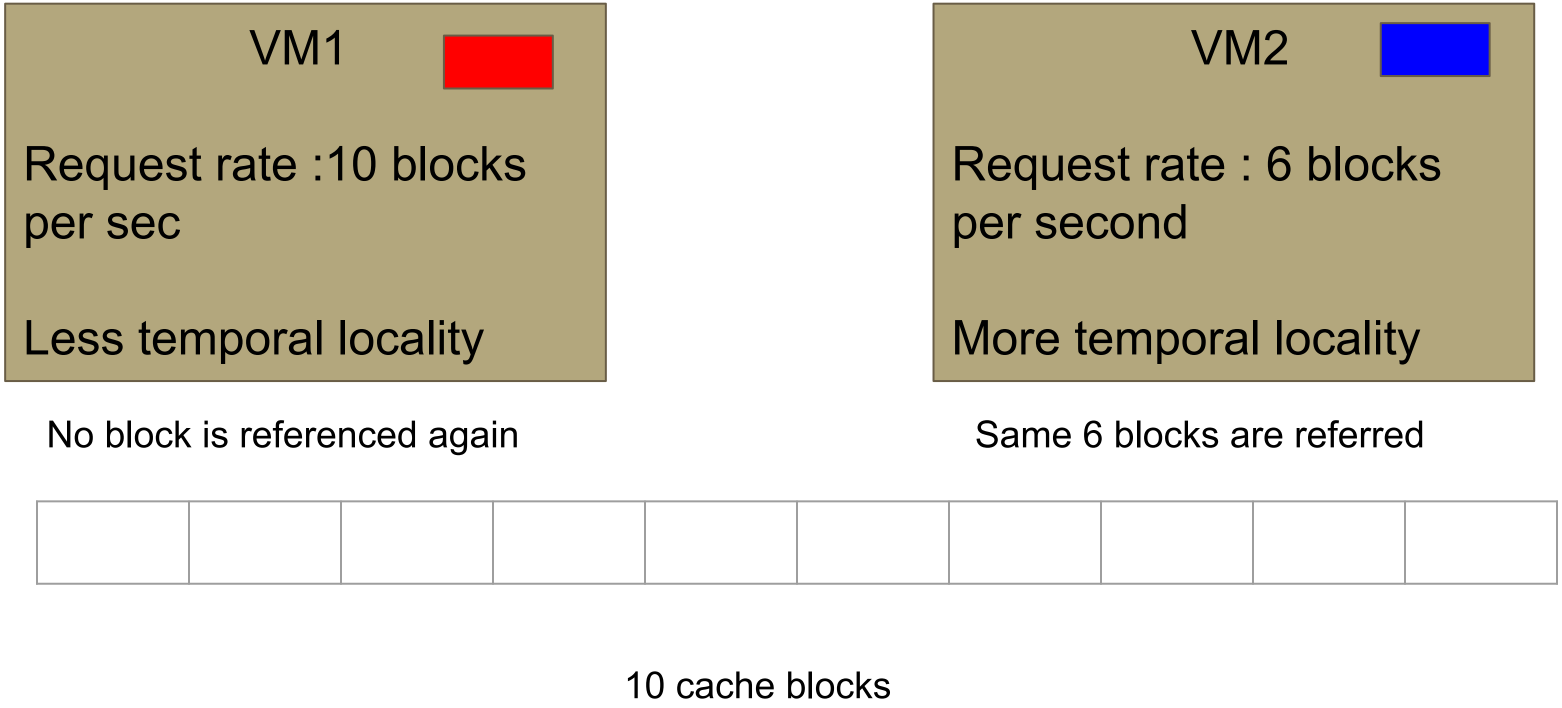
Challenges

- ❏ Consistency
- ❏ Cache eviction policy
- ❏ Cache sizing

Types of caching



Why cache partitioning is required?



Why cache partitioning is required?

VM1

Request rate :10 blocks per sec

Less temporal locality

VM2

Request rate : 6 blocks per second

More temporal locality

1	2	3	4	5	6	7	8	9	10

10 cache blocks

Why cache partitioning is required?

VM1

Request rate :10 blocks per sec

Less temporal locality

VM2

Request rate : 6 blocks per second

More temporal locality

100	101	102	103	104	105	7	8	9	10

10 cache blocks

Why cache partitioning is required?

VM1

Request rate :10 blocks per sec

Less temporal locality

VM2

Request rate : 6 blocks per second

More temporal locality

11	12	13	14	15	16	17	18	19	20

10 cache blocks

Why cache partitioning is required?

VM1

Request rate :10 blocks per sec

Less temporal locality

VM2

Request rate : 6 blocks per second

More temporal locality

100	101	102	103	104	105	17	18	19	20

10 cache blocks

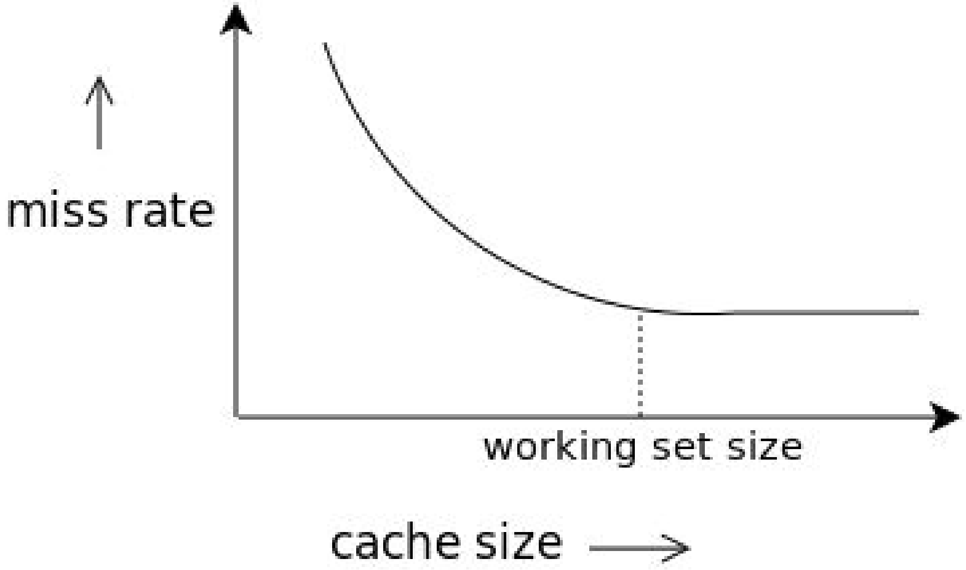
Single level caching - with dynamic partitioning

For each VM collect IO traces

<vmid, block address, time>

For each VM construct MRC

Miss rate curve



- Direct method
- By reuse distance algorithm
- By cache miss equation

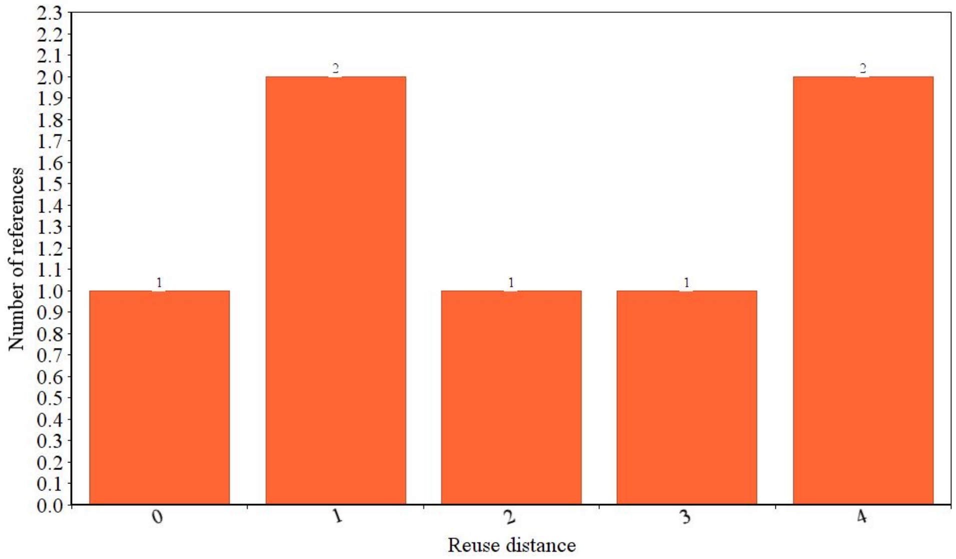
Reuse distance algorithm

Reuse distance of reference 'B' is number of unique references since the previous access to 'B'

time	0	1	2	3	4	5	6	7	8	9	10	11
references	d	a	c	b	c	c	e	b	a	d	a	c
Reuse distance	infi	infi	infi	infi	1	0	infi	2	3	4	1	4

Reuse distance algorithm

Hits = Sum(number of references) whose reuse distance < cache size



Cache size	hits	Misses (12 - hits)
1	1	11
2	3	9
3	4	8
4	5	7
5	7	5
6	7	5

Cache miss equation

$$Missrate = \frac{1}{2}(H + \sqrt{H^2 - 4})(P^* + P_c) - P_c$$

$$where: H = 1 + \frac{M^* + M_b}{M + M_b} for M \leq M^*$$

P^* : Minimum miss rate possible, i.e. miss rate at cache size = working set size
M^* : Cache size beyond which miss rate won't decrease
M_b : Memory needed by cache manager for storing metadata information
P_c : Miss rate curve's convexity

Metric used for partitioning

For each VM i , calculate

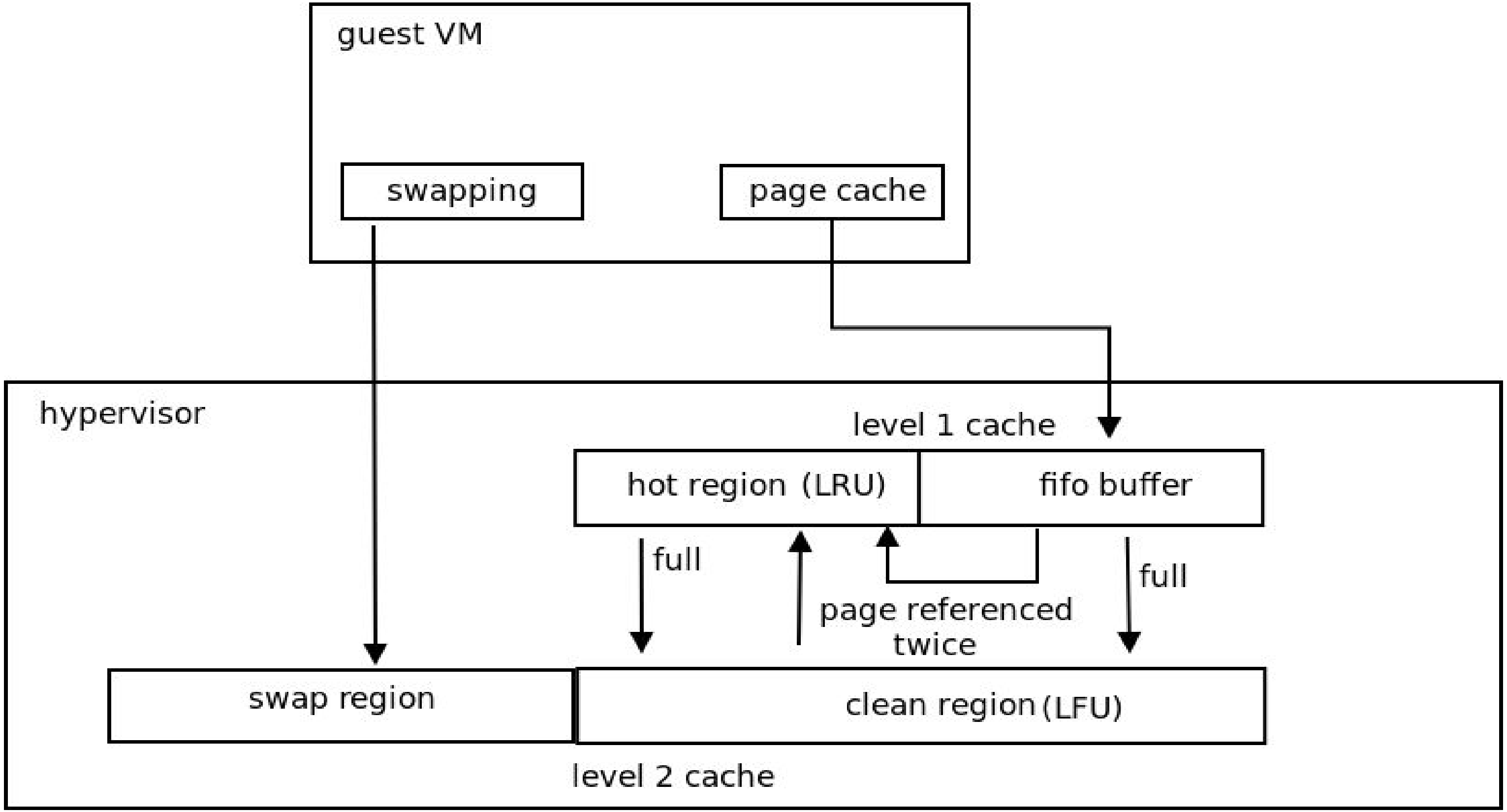
Cache utility (Ci) = cache hit ratio (depends on cache size, Si) x device latency

use probabilistic search to find Si such that :

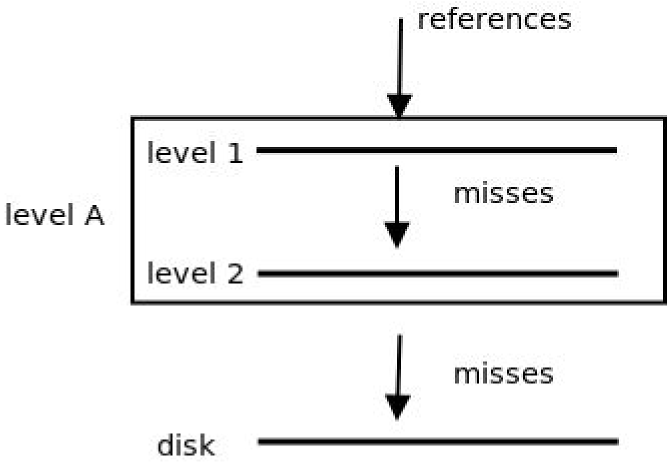
$$\Sigma C_i(S_i) \times P_i = \text{maximized}$$

$$\Sigma S_i = \text{Total cache size}$$

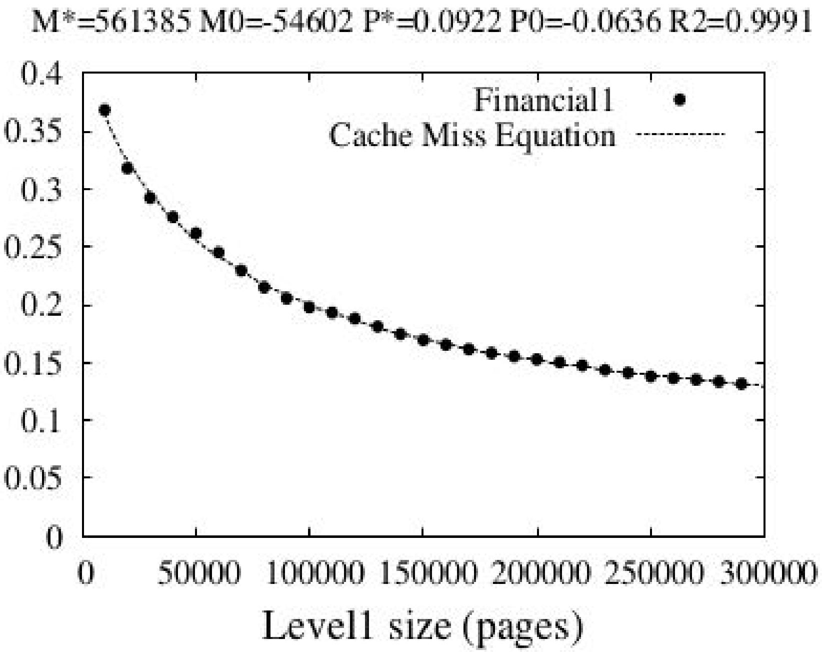
Multilevel Caching - no per VM partitioning



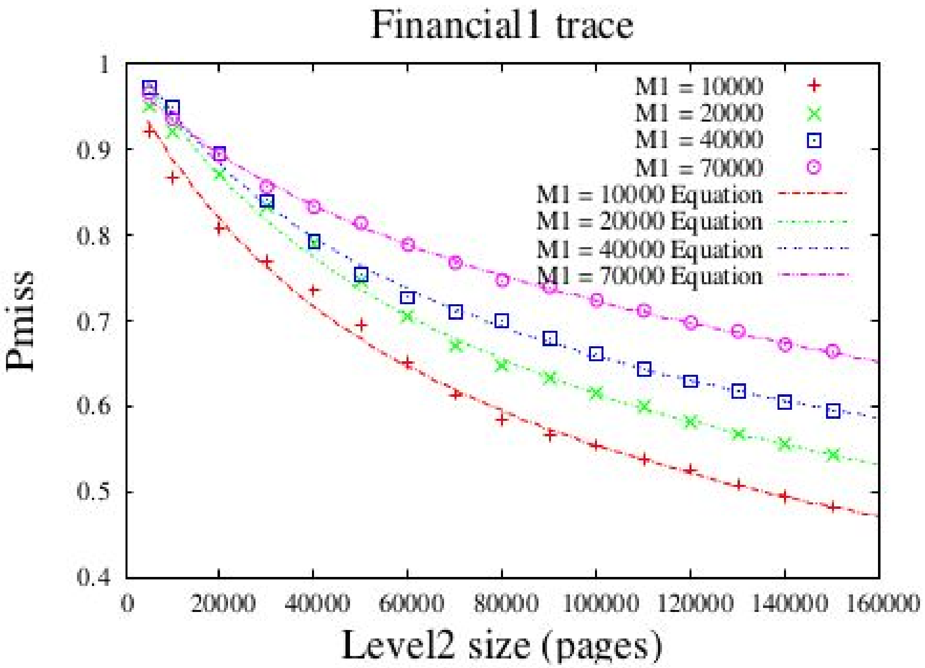
Multilevel Caching - with partitioning



MRC of level 1 and level 2

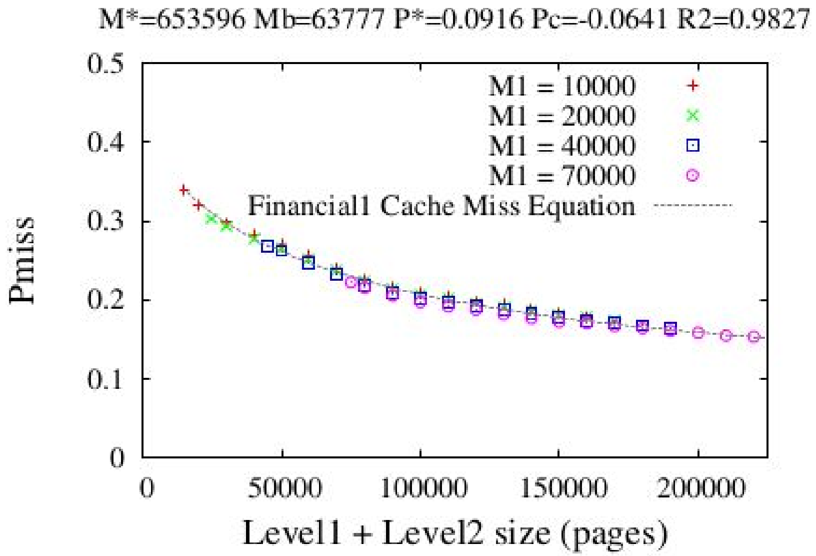


(a) Level_1



(b) Level_2 for different M_1 values.

MRC of level a



(c) Level_a: Level_1 plus Level_2.

Determine size of level 1 partition

Level 1 cache get the partition size for fair allocation

$$S_i^1 = \beta_i(M^1 + \sum_{r=1}^{k} M_{br}^1) - M_{bi}^1$$

$$where: \beta_i = \frac{(\frac{P_{ci}^1}{P_i^{1*}} + 1)(M_i^{1*} + M_{bi}^1)}{\sum_{r=1}^{k}(\frac{P_{cr}^1}{P_r^{1*}} + 1)(M_r^{1*} + M_{br}^1)}$$

Parameters are same as that of cache miss equation:
S_i^1 : $level_1$ cache size of VM_i
S_i^2 : $level_2$ cache size of VM_i
M^1 : Total cache size at $level_1$
M^2 : Total cache size at $level_2$
P_i^{1*} : Minimum miss rate possible of $level_1$ for VM_i
M_i^{1*} : Cache size of $level_1$ for VM_i beyond which miss rate won't decrease
M_{bi}^1 : Memory needed by cache manager for storing metadata information of $level_1$ for VM_i
P_{ci}^1 : Miss rate curve's convexity of $level_1$ for VM_i

Determine size of level 2 partition

Service level objective : Level a

$$Prob(total\ latency > L_{max}) < p_{max}$$ --------------------------(1)

$$\begin{aligned} Prob(T > L_{max}) = {} & Prob(x \in level_1)Prob(L_1 > L_{max}) + \\ & Prob(x \notin level_1\ and\ x \in level_2)Prob(L_2 > L_{max}) + \\ & Prob(x \notin level_1\ and\ x \notin level_2)Prob(L_3 > L_{max}) \end{aligned}$$

$$Prob(T > L_{max}) = P_a^{miss} Prob(L_3 > L_{max})$$ --------------------------(2)

$$Miss\ rate\ at\ level_a = \frac{0.95 p_{max}}{Prob(disk\ latency > L_{max})}$$

Partition size at level 2 = Level a - Level 1

Summary

Need to improve disk io latency

Different ways of caching in virtualized systems

- Single level caching with partitioning
- Multilevel caching with and without partitioning

Why partitioning is better than unified cache

Cache eviction policy

Possible future work

- Comparative analysis of all the caching techniques

- Implement dynamic per VM cache eviction policy

References

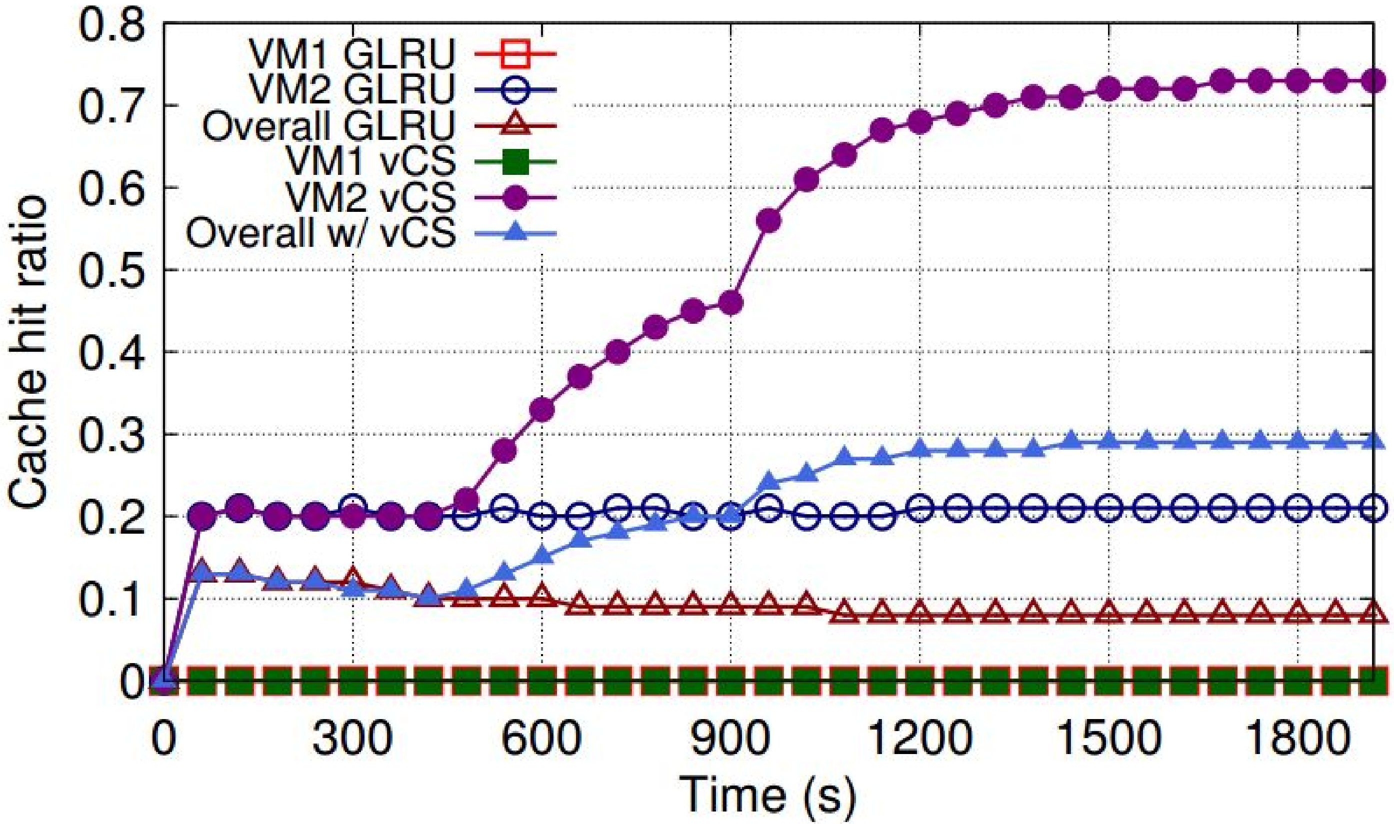
- Meng Fei, Zhou Li, Ma Xiaosong, Uttamchandani Sandeep, and Liu Deng. vCacheShare : Automated Server Flash Cache Space Management in a Virtualization Environment. Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference, pages 133–144, 2014.
- R. Koller A. J. Mashtizadeh R. Rangaswami. Centaur: HostSide SSD Caching for Storage Performance Control. Proceedings of Autonomic Computing, 2015 IEEE International Conference on cloud computing, pages 966–973, 2015.
- Y. C. Tay and M.Zou. A page fault equation for modeling effect of the memory size. Proceedings of Perform Eval, pages 99–130, 2006.

References

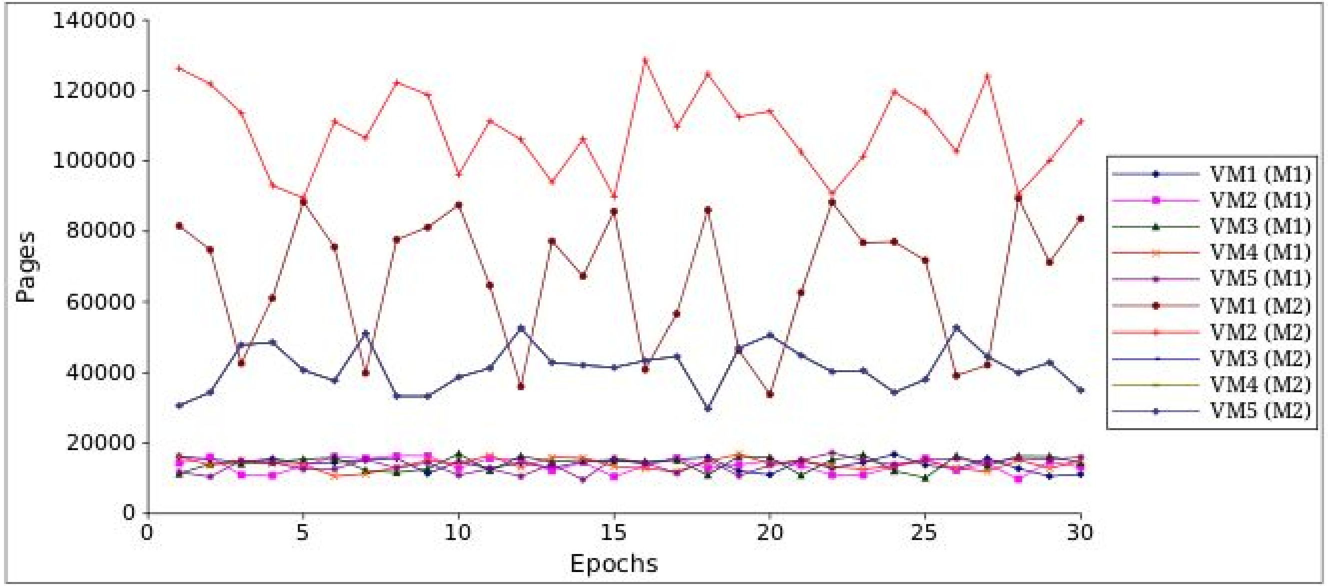
- Vimalraj Venkatesan, Wei Qingsong, and Y. C. Tay. ExTmem: Extending Transcendent Memory with Nonvolatile Memory for Virtual Machines. Proceedings of the 2014 IEEE Intl Conf on High Performance Computing and Communications, pages 51–60, 2014.
- Vimalraj Venkatesan, Wei Qingsong, Y. C. Tay, and Yi Irvette Zhang. A 3level Cache Miss Model for a Nonvolatile Extension to Transcendent Memory . Proceedings of the 6th International Conference on Cloud Computing Technology and Science, pages 218–225, 2014.

Thank you

Experiment - Unified vs partitioned



Experiment - Dynamic partitioning



(a) Dynamic allocation of M_1 and M_2 to 5 VMs.