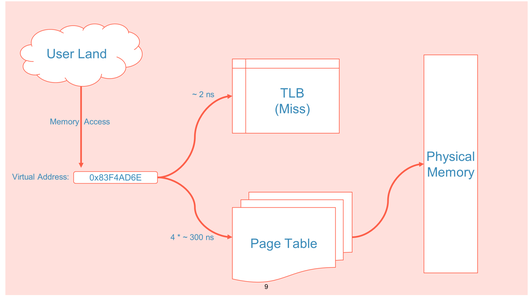
Tip of the Iceberg: Low Associativity Paging

Nirjhar Mukherjee

The TLB Coverage Problem



- Traditional fully-associative page mapping is slow.
- TLBs are designed to solve this issue by caching mappings.
- Unfortunately, TLB coverage is often poor, resulting in frequent misses.
- Increasing coverage may help → but TLB size grow slowly.

Ways to reduce associativity

Hashing is the preferred mechanism to reduce associativity.

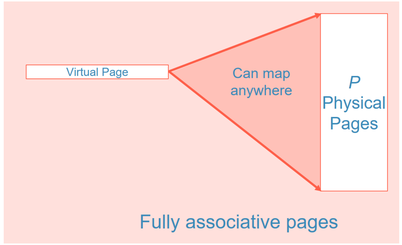
Most commonly used hashes include left[*d*] and single.

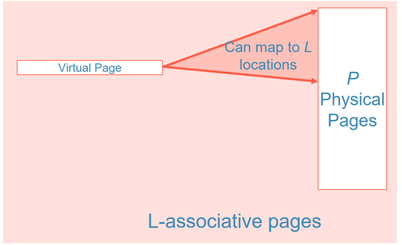
However they both have flaws:

- Left[*d*]'s bound don't hold under dynamic load
- Single has high variance, resulting in fragmentation

Fully Associative Page mapping allows a virtual address to be mapped to any physical address

L-Associativity Page mapping allows a virtual address to be mapped to a set of L physical address.





Associativity can be leveraged to increase TLB coverage.

Lower associativity requisites fewer bits to represent a mapping.

This means more mapping can be stored in a fixed amount of TLB memory! In fact, lower the L, the fewer the number of bits required to represent a mapping!!

A solution

Iceberg[*d*] Hashing uses a mix of the two:

- It uses single for most mappings
- And uses Left[*d*] for a small number of mappings

This can increase TLB coverage by up to 8x

Implementing and tuning in the kernel

One caveat of hash based low associativity page mapping schemes is the existence of associativity conflicts: What happens when a virtual page is hashed to a set of L pages, which are all already mapped?

We tune iceberg with various sizes of L to find a good fit – Minimize L (or rather b as in bucket (of pages) size), while also minimizing associativity conflicts

Sweet Spot

	Associativity Conflicts for Fraction of Memory Allocated			
b	85%	90%	95%	100%
8	0	4 (0.01%)	266 (0.47%)	2723 (4.84%)
16	0	0 (0.00%)	35 (0.06%)	1720 (3.05%)
32	0	0 (0.00%)	0 (0.00%)	867 (1.54%)
64	0	0 (0.00%)	0 (0.00%)	639 (1.13%)
128	0	0 (0.00%)	0 (0.00%)	391 (0.69%)

Another Aspect of this problem is finding the right iceberg parameter *d* (magically, optimal bucket size due to d is also 32) and dealing with associativity conflicts (using LRU). This discussion is elaborated upon in the thesis.

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL