

COMP9242 Paper 2 Review

Z5141448

August 24, 2019

1 Summary

This paper talks about the performance issue of using Huge Page in Linux and trying to address that with a new method. The Huge Page mechanism is to reduce the Translation Look-aside Buffer (TLB) misses increasing the performance. While using TLB, there's a new problem of memory bloating and low utilization of huge pages. So the huge must be split and re-merge together in some scenario.

The traditional method in Linux is using the access bit in the Page Table Entry (PTE), and sampling overhead achieves 6 seconds as claimed. The new method uses the TLB misses by using SysMon-H with a shadowed array to record the TLB misses address at run-time, which reduces the overhead.

Also, the paper introduces the H-Policy, which delays the splitting and merging of Huge Page and small page to improve the performance. The H-Policy uses the data provided by SysMon-H and a sorted Huge page tree (H-Tree) to improve the lookup of the cold page and split them.

2 Pros

- There's a potential speed up if the mechanism is well implemented
- The paper is very concise

3 Cons

- The benchmark has no further information, potential benchmark crime
- Key assumption does not address
- Unable to reproduce their work

4 Criticism of the work

4.1 Trustworthiness of the Paper

Firstly, it's potential a Benchmark Crime. It said "overhead achieves 6 seconds in a specific sampling window" which has not to specify how long the sampling window, standard deviation and in what trigger the system too frequently checking the access-bit. When I first see the 6 seconds, I shocked, but after I review the statement, I found it's not that trusted. Also, the overhead of 2.3% and vs 0.1% in the new method has not further clarification.

Secondly, it claimed the H-Page would reduce the access time, which is true, but it doesn't reduce that much, the overall sorting complexity of sort tree and an array are both $O(n \log n)$.

Thirdly, the TLB misses record only assess 5 seconds for a period while the access-bit is recorded continually, which makes the SysMon-H method has some advantage. If the TLB misses record continually, it may not have performance improvement.

Fourthly, the citation is highly irrelevant, for example, 22, 27, 2... and use Wikipedia in citation such as 5 and 6.

Last but not least, I couldn't find any material about the essential component of SysMon-H, H-Policy and H-Tree mentioned in the article, which means there's no way for any others to reproduce their work.

4.2 Missing Support for Key Assumption

There's a hidden assumption "hot pages usually incur a large number of TLB misses", I feel the statement is true, but there's no further clarification or any numbers to support this statement while the whole article built on top of this statement. It's trivial to compare the access-bit in the hot region as they can open both the feature. After all, they are two different kinds of things, and if the author wants to use one instead of others, it's better to have a good reason.

4.3 Not Deep Enough Analyze

The article addresses the problem with high memory-intensive programs, and there is discrete use of memory and usually doesn't need the THP [1]. Maybe for those applications in the real world need to open the Huge Page and no merging, which give better performance. Because there's no need to compact the pages on air which hurts performance badly, also the splitting of Huge Page may cause more TLB misses which leads to that page unable to merge back together?

There's missing performance analyze of H-Policy vs original THP policy. There should be a comparison between the new method and the old method. The lazy splitting should have some benefit of performance but maybe not that promising. As my point of view, there's not technical advance in H-Policy so the improvement may be little.

4.4 Shadow Array Issue

What's the implementation of shadow array, how can it hold all the TLB fault address in the exam period while not hurting the performance? If the array is running out, what will happen? There's no information about this issue.

5 Points that Must be Addressed if Accepted

"hot pages usually incur a large number of TLB misses", and the relationship between the access bit and the TLB misses.

Clarify the performance of H-Policy vs original THP policy.

Provide More information about SysMon-H and the shadow array.

References

- [1] Disable Transparent Huge Pages (THP) MongoDB Manual
<https://docs.mongodb.com/manual/tutorial/transparent-huge-pages/>