

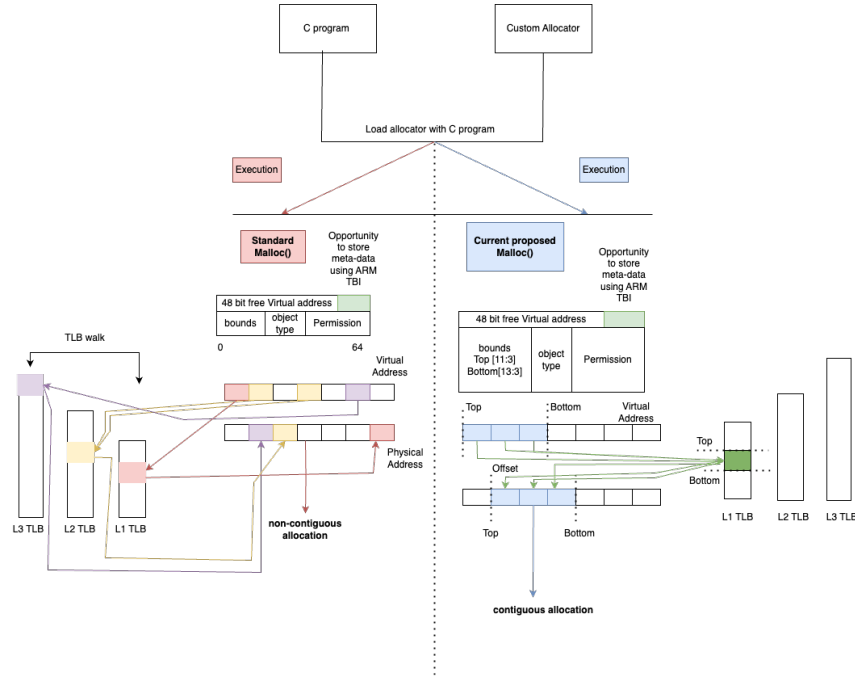
Contents

Future work

This documents is decision making to highlight potential paths to take for this PhD. We will initially talk about the current expirement which is a FAT pointer based memory allocator and will then expand into 2 potential paths:

- Cheri RISCv to prevent using the TLB.
- Allocator evaluation based on stripping instruction calls for larger allocators like Jemalloc.

1. Current expirement: FAT pointer based range addresses



The objective of this expirement was to ensure we can use the CHERI bounds as tracking mechanism of allocations instead of using multiple TLB entries. Using this approach we can use a single Huge page entry with bounds to ensure that the bounds (Which is the top and base address) can be extracted from the pointer using the Cheri compressed bounds mechanism.

We implemented a simple allocator which uses this technique with a basic malloc and free.

Objectives

- How does the utilization of bounds for tracking memory allocations, in addition to security purposes, affect the run times and Translation Lookaside Buffer (TLB) miss rates in modern computing systems ?
- How does the implementation of bounds for seeking through physically contiguous memory influence the complexity and efficiency of standard memory allocators, particularly those with advanced features such as transparent huge pages, and what are the implications for system performance in terms of execution speed, memory access latency, and resource utilization?

Hardware

- ARM morello (Huge page size 1GB used)

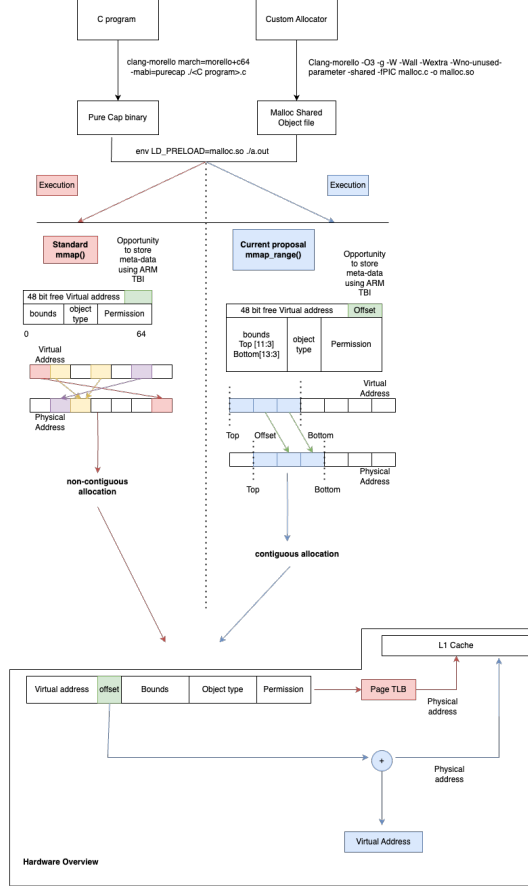
Evaluation

We conducted tests of the FAT Pointer-based range addresses against Jemalloc, the default memory allocator for CHERI BSD, to assess the performance improvements enabled by a CHERI-based huge page-aware allocator. Specifically, we evaluated the reduction in TLB misses and its impact on overall performance metrics, such as wall clock runtime. To comprehensively analyze the proposed allocator, we categorized benchmarks into two classes which are micro and macro benchmarks. Micro benchmarks comprise smaller C programs designed to target specific allocator patterns, enabling us to evaluate detailed aspects of the allocators behavior. Macro benchmarks, on the other hand, encompass larger, realworld C programs, allowing us to assess the allocators performance in more practical, real-world scenarios.

limitation

- Using Huge page still requires a TLB entry which could be mitigated (Refer to the FPGA work).
- ARMv8 only supports using to virtual addresses so it's required to bypass the TLB for address translation.

2. Cheri RISC-V to prevent using the TLB



In the current ARM Morello setup, address translations rely on the TLB. The future approach on RISC-V Tooba involves storing the offset directly within the pointer. This is possible due to Cheri's capability model, which supports fine-grained memory protection and can encode bounds within pointers. Utilizing Bounds in Cheri for Block-Based Allocation: Cheri capabilities allow pointers to carry metadata about memory bounds, providing hardware-enforced memory safety. By encoding the offset and bounds within the pointer, the system can directly access memory without needing intermediate translations via the TLB. This enables the implementation of a block-based allocator that can efficiently manage memory allocations and deallocations within defined bounds. Bypassing the TLB in RISC-V Tooba.

Hardware modifications

The Bluespec design of the RISC-V processor will be modified to allow certain memory operations to bypass the TLB. This means that when a pointer with encoded offset and bounds is used, the system can directly compute the physical address from the capability information. This modification reduces the dependency on the TLB, decreasing latency. and improving performance, especially for frequent memory operations.

3. Allocator evaluation based on stripping instruction calls for larger allocators

