Memory Virtualization - Recitation

Rishabh Iyer

28. 10. 2021

Lecture Recap

- Would we need VM if we had unlimited physical mem?
    - Which entity enforces memory isolation between processes?
- Why are page tables hierarchical in x86?
- What is the minimum page size in x86? Why?
    - How does this impact page table organization?
- Why must TLB hit rates be >=95%?
- How are the L1/L2/L3 caches in Skylake indexed,? Why?

Agenda for today’s recitation

- Recap CHERI paper

- Design exercise based on CHERI

Main principles underlying CHERI

- Principle of least privilege
  - A subject should be given only those privileges needed for it to complete its task
- De-conflating virtualization and protection
  - Paging is best-suited for translation, segmentation for protection.

Capabilities

- Capabilities are an unforgeable token of authority
  - Enable de-centralized, minimal overhead access control
- CHERI capabilities are unforgeable fat pointers
  - Fat pointer = Base + Length + Permissions

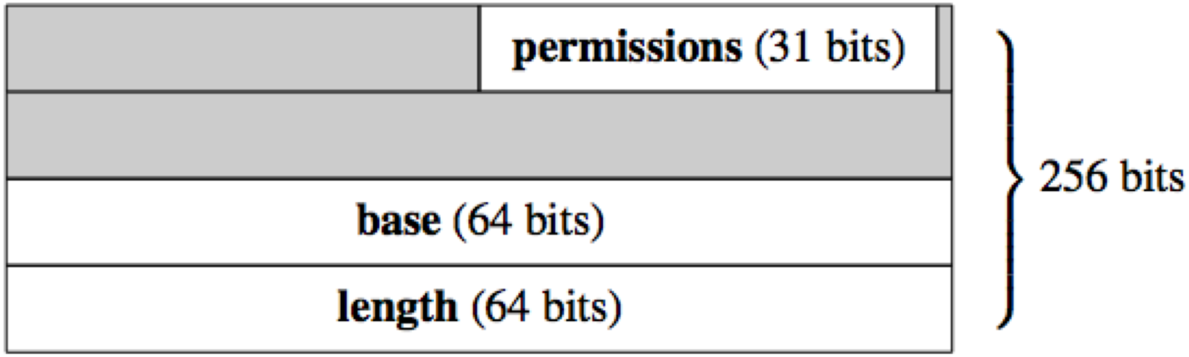


Figure 1: Memory capability

Memory Safety using capabilities

- How do capabilities ensure memory safety?
- What defines the current protection domain?

Design goals for a RISC capability system

- Capability manipulation must be unprivileged
- Capabilities can span any range in the VA space
- Should be able to run unmodified legacy code, while still being constrained by capabilities

Capabilities in CHERI

- Where are they stored?
- Assuming the right capabilities are present, how are they enforced?

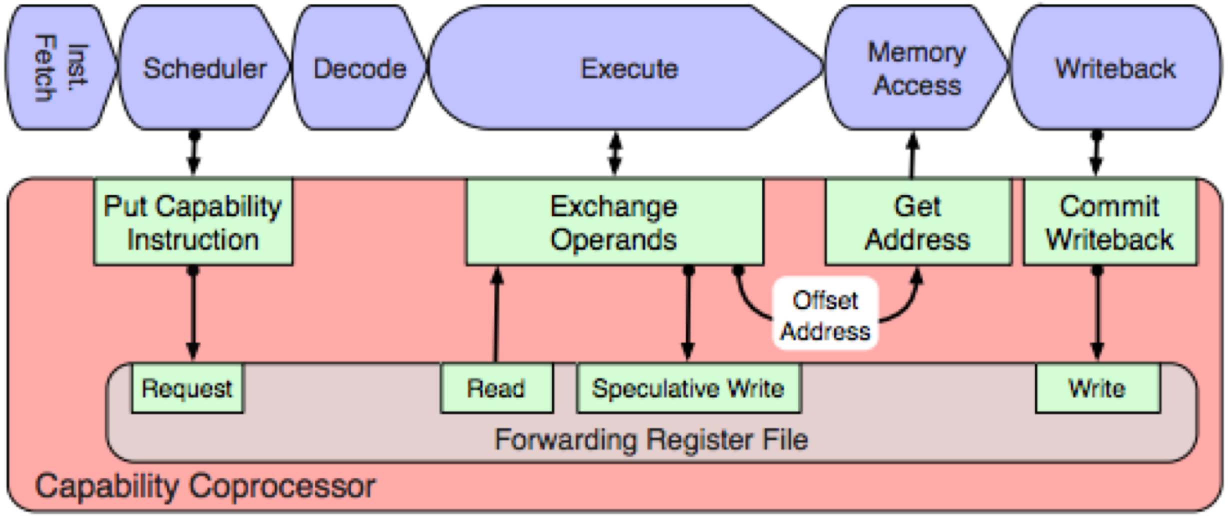
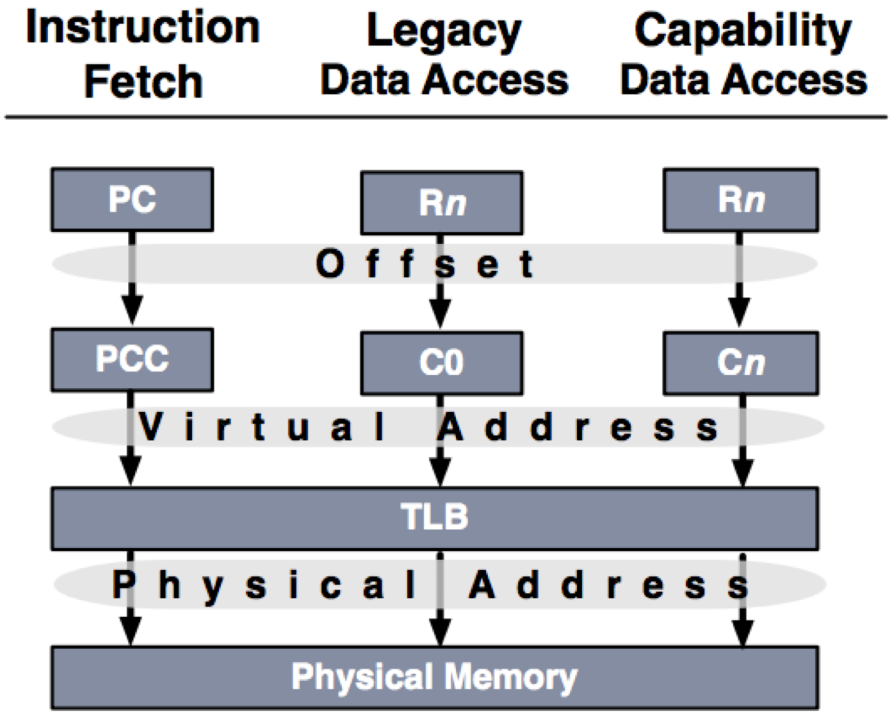


Figure 2: BERI pipeline with capability coprocessor

Support for legacy code in CHERI

- How does legacy code fit into the capability model?
- Does CHERI make legacy code memory safe?



Ensuring relevant capabilities

- What role do they OS, compiler and HW play?
- Consider the following example:
  - ❖ I malloc an int. The kernel mmaps a page of memory with VA 0 to 4KB. libc then returns allocates the integer in the first 8 bytes.
  - ❖ I now malloc a second int. libc does not need to call mmap(), it simply allocates the integer in the next 8 bytes of the page above.
  - ❖ I now want to use these two integers, how does the hardware have the correct capabilities?

Ensuring relevant capabilities

- OS creates capabilities for process
  - Done when process is loaded/new memory is allocated
- Compiler associates each load/store with the correct capability register
  - Ensures that the hardware does not need to check the entire capability table
- Hardware ensures unforgeability of capabilities
  - Only provides support to **reduce** privileges
  - Ensures regular stores cannot modify capabilities using tags

Design Exercise: Compartmentalization

- Design an efficient application-level compartmentalization scheme within CHERI to ensure isolation of untrusted, third party code. Your scheme must allow for the isolated code to be given arbitrary (but defined) access to the virtual address space. You may assume compiler support when invoking the third party code.