guest_memfd in-place sharing and 1G page support

updates and questions

ackerleytng, 2025-04-03

Update

- Using HugeTLB to provide 1G page support in guest_memfd
- Building off Fuad's two series [1] [2]
- Still working out refcounting and locking issues

[1] https://lore.kernel.org/all/20250318161823.4005529-1-tabba@google.com/T/
[2] https://lore.kernel.org/all/20250328153133.3504118-1-tabba@google.com/T/

Q: Transient refcounts on guest_memfd pages

Is it okay for guest_memfd to fail conversion if there are elevated refcounts?

Q: Transient refcounts on guest_memfd pages

Current proposal in Fuad's series v7 [1] and how folio_put() callback is used

- If refcount == safe refcount
  - Set folio (technically offset) to `KVM_GMEM_GUEST_SHARED`
- Else
  - Disallow faulting by setting folio to `KVM_GMEM_NONE_SHARED`
  - Setup folio_put() callback
    - folio_put() callback is guest_memfd's notifier that there are no more users, then we can set folio to `KVM_GMEM_GUEST_SHARED`
- If guest tries to fault memory in `KVM_GMEM_NONE_SHARED` state vcpu_run() returns `-EBUSY`, userspace retries

[1] https://lore.kernel.org/all/20250328153133.3504118-1-tabba@google.com/T/

Q: Transient refcounts on guest_memfd pages

Alternative proposal for conversion

- If refcount == safe refcount
  - Set folio (technically offset) to `KVM_GMEM_GUEST_SHARED`
- Else
  - Return `-EAGAIN` (“Resource temporarily unavailable”) to guest_memfd’s caller
  - For X86, `-EGAIN` will go out to userspace, userspace can try again, and the possible reasons for getting this error are that
    - Userspace VMM forgot to unpin one of the pages
    - There’s a transient refcount on one of the pages in the requested conversion range (which should not happen often) and userspace should retry

Q: Transient refcounts on guest_memfd pages

Why?

- More transparent errors
  - Userspace, guest, or someone must retry while the page conversion hasn't completed
  - Deferring the retry till when guest tries to fault in the page and sees `KVM_GMEM_NONE_SHARED` is less obvious than an error during conversion
- Removes the need for third `KVM_GMEM_NONE_SHARED` state
- Removes(?) the need for folio_put() callback
- DavidH: returning error is an interface that is more reversible (can reduce errors later). The opposite interface is harder to change.
- DavidH: can perhaps skip speculative refcounts for guest_memfd folios in future

Thanks :)

Q2: guestmem (the library) interface to split and merge?

Context

- Splitting and merging folios takes significant amounts of time
  - Want to be able to control when it happens, as opposed to leaving it up to when the last refcount is dropped (more uncertainty and possible random latencies)
- Splitting is a requirement
  - For per-page pincount and refcount tracking
  - Because core-mm doesn't support 1G page mappings outside of HugeTLB (yet)
- Merging when converting to private is an optimization
  - Might be worth avoiding the merge if guests keep converting back and forth
- Userspace can apply heuristics on
  - When to merge, what size to merge to - 2M or 1G, and what size to split to, etc
  - James Houghton: Why not merge only at the end?
  - Michael Roth: SNP cannot merge pages unless guests request it. If we're optimizing there, why not optimize splitting?
    - Vishal: Will still save memory from HVO
  - DavidH: Maybe an guest_memfd ioctl to merge?
    - HugeTLB: Buddy split patch series
  - DavidH: focus on using folio_put() callback to merge only at truncate
  - DavidH: What to do on reboot? Request merge on reboot?