guest_memfd upstream call: Circling back on memory failure

For 2025-08-07 guest_memfd bi-weekly upstream call
Contact ackerleytng@google.com if you have questions/suggestions!

Action/Component	**Upstream, initial stage**	**Upstream, eventually**
Unmap from host page tables	Entire file worth	Split and unmap failed page within folio
Unmap from stage2 page tables	Entire file worth	Split and unmap failed page within folio
Mark folios HWpoisoned on handling #MCE	Yes	
Track PFN that failed	With HWpoison flag	
Indicate memory failure when folio is returned to HugeTLB	With HWpoison flag	
Error on faulting **private** memory with failure	KVM exit with -EHWPOISON	
Error on faulting **shared** memory with failure	SIGBUS	
Additional work HWpoison during folio restructuring	None, since guest_memfd is not functional upon memory failure	Distribute HWpoison flag during split Disable merging during conversions when some part of a HugeTLB folio is poisoned. Implement summarizing of the HWpoison flag during merge just for returning the page to HugeTLB.
`PR_MCE_KILL_EARLY` handling	If set, SIGBUS while handling #MCE	
Signal userspace while handling #MCE	If folio is dirty and folio is faulted into host userspace, SIGBUS all the processes where the memory is currently mapped in host page tables. In addition, SIGBUS `current` for consumed memory failure, no signal for deferred	

	memory failure.

Discussion Points

- Now, guest_memfd always sends SIGBUS to `current`, including for **deferred** memory failures? Bug or feature?
  - `kvm_gmem_error_folio()` does not truncate the folio and returns `MF_DELAYED`
  - Caller, `me_pagecache_clean()` sees elevated refcounts (since the folio was not truncated) and returns `MF_FAILED`
  - `memory_failure()` will return `-EBUSY`, leading to `kill_me_now()`
    - James: `-EBUSY` leads to `kill_me_now()` is weird
    - Sean: Do the same as any other filesystems
- For the initial stage, we discussed SIGKILL-ing `current` at the last meeting. Is that sufficient?
  - A second process that has the bad memory mapped can still access bad memory.
  - James: probably should unmap
    - Consuming a few times is okay?
  - Dan: Seems reasonable to delay, but people might hate seeing multiple #MCEs
    - Dax already did that work, look into that.