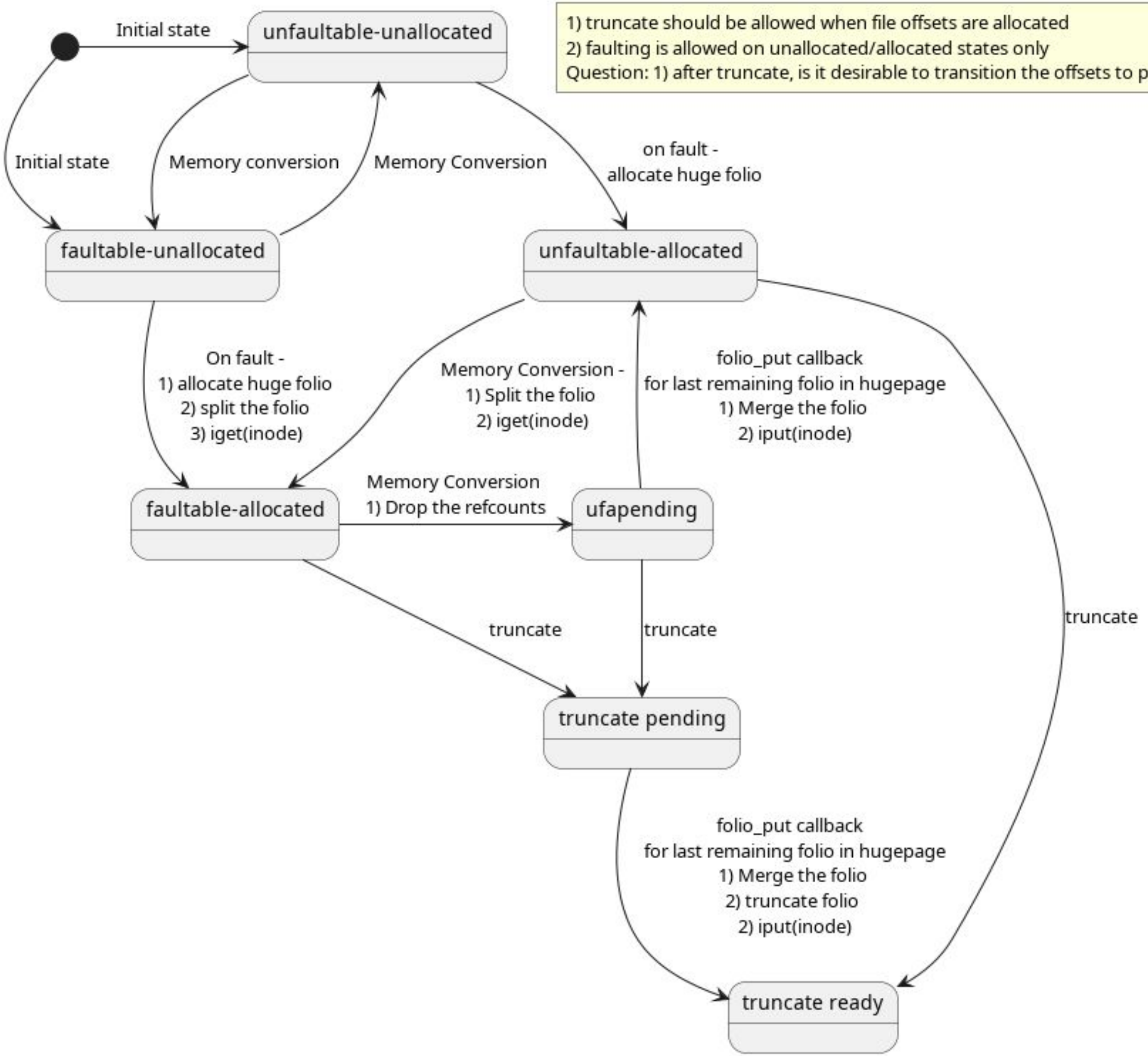
Guest Memfd: Hugepage Support

vannapurve@

Guest_memfd: Huge Page Folio Support

- Faultable ranges are backed by split folios.
- It should be safe to merge the page back when complete hugepage range becomes unfaultable or is truncated.
- Guest_memfd needs to return folios to the allocator as they were allocated.

1) truncate should be allowed when file offsets are allocated
2) faulting is allowed on unallocated/allocated states only
Question: 1) after truncate, is it desirable to transition the offsets to previous faultable/unfaultable states?



Initial state

unfaultable-unallocated

Initial state

Memory conversion

Memory Conversion

on fault -
allocate huge folio

faultable-unallocated

unfaultable-allocated

On fault -
1) allocate huge folio
2) split the folio
3) iget(inode)

Memory Conversion -
1) Split the folio
2) iget(inode)

folio_put callback
for last remaining folio in hugepage
1) Merge the folio
2) iput(inode)

Memory Conversion
1) Drop the refcounts

faultable-allocated

ufapending

truncate

truncate

truncate

truncate pending

folio_put callback
for last remaining folio in hugepage
1) Merge the folio
2) truncate folio
2) iput(inode)

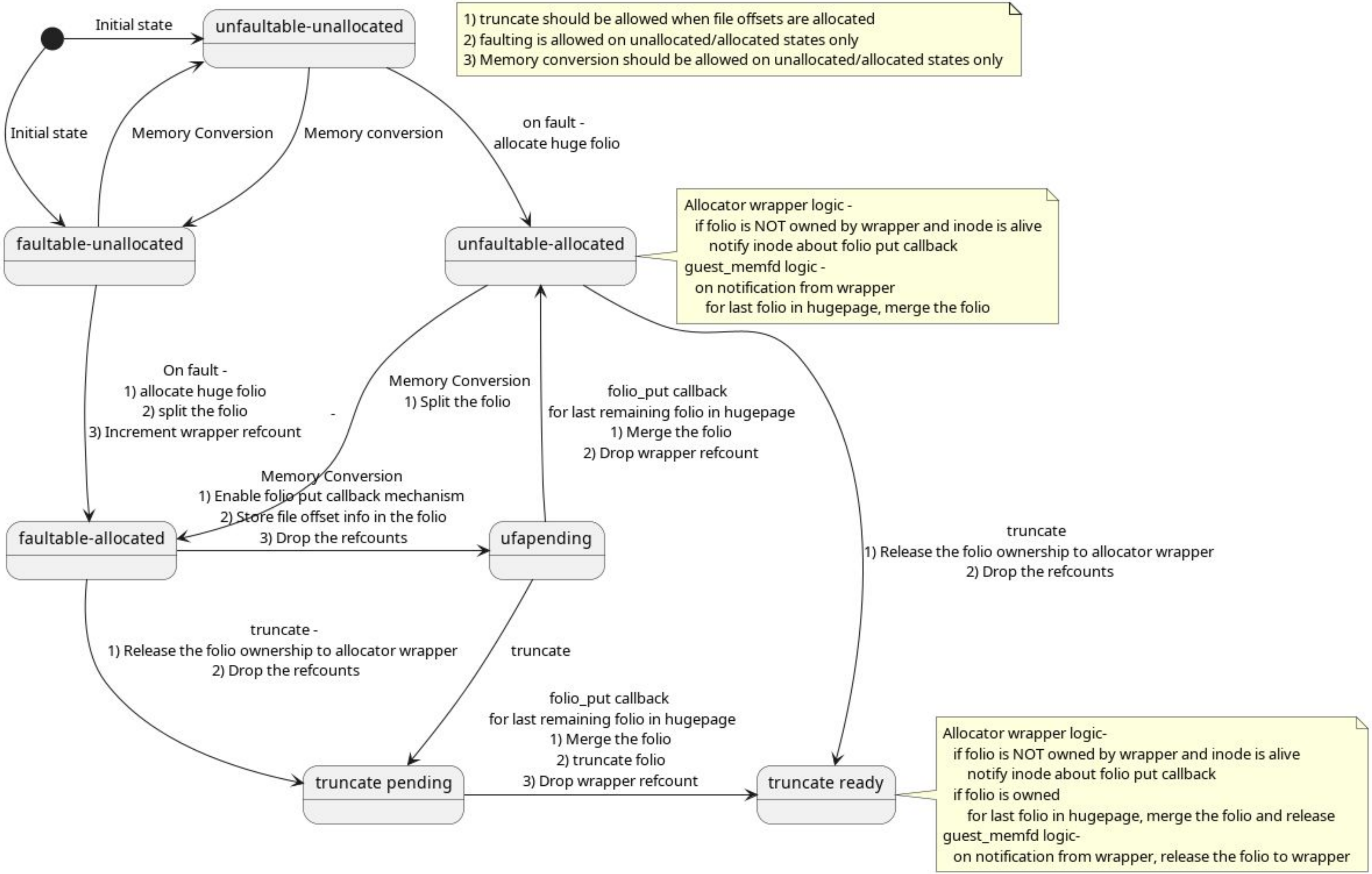
truncate ready

Issue: Does it make sense to introduce fapending?

- Deferring split of the folio till shared faults happen, can cause guest_memfd users to wrongly think that the folio can still be mapped at larger size.
- Better would be to implement eager split during memory conversion.

Issue: Restoring folios before vs after inode cleanup

- Keeping inode alive till all folios are returned to their unsplit state is problematic.
  - Inode truncate will not get triggered unless userspace explicitly does so.
- Option: Introduce an allocator wrapper which can stick around till all the folios are restored.
  - Can be implemented as part of mm/guest_memfd library.
  - Should handle folio_put callback and relay it to guest_memfd if needed.



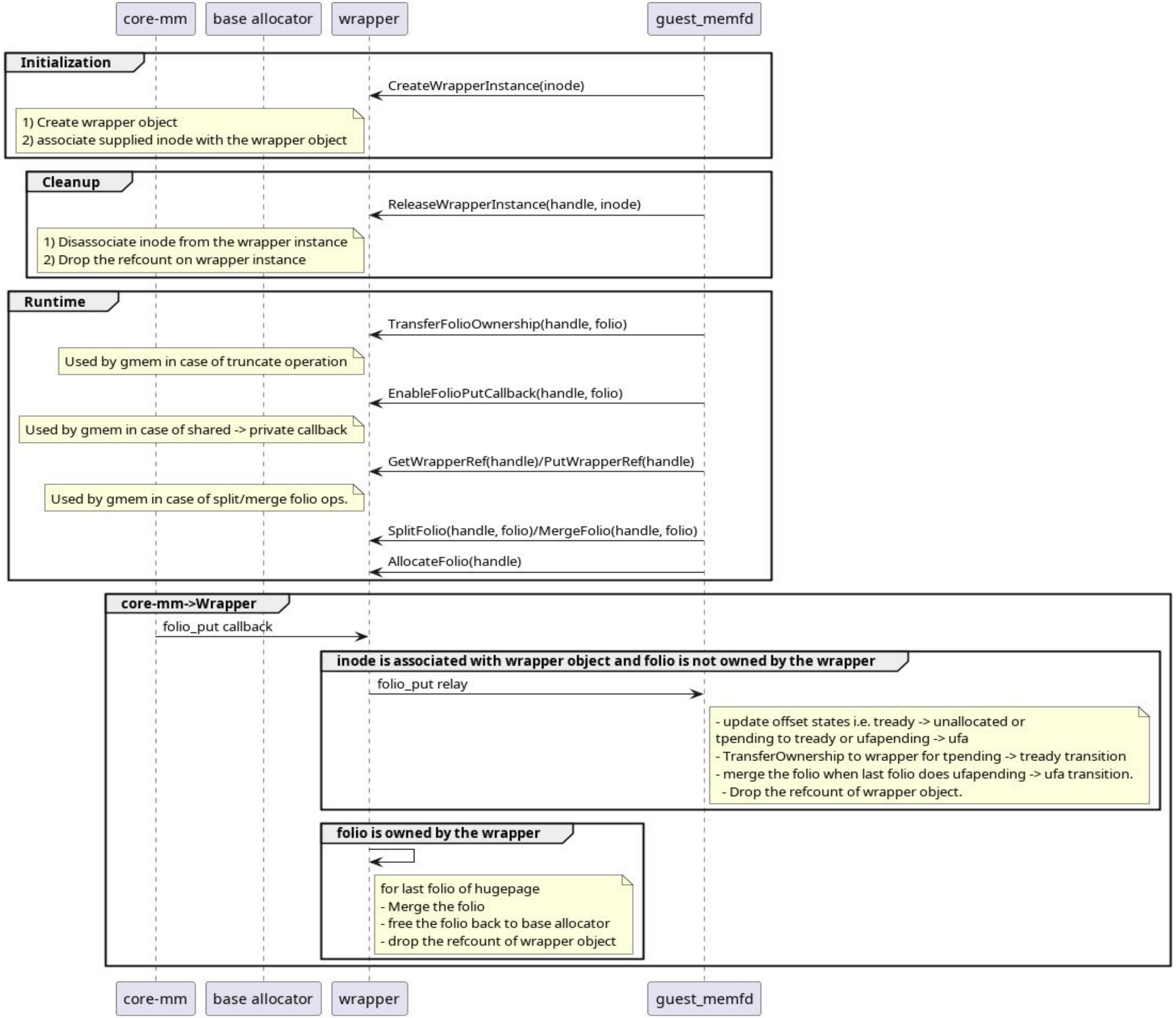
1) truncate should be allowed when file offsets are allocated
2) faulting is allowed on unallocated/allocated states only
3) Memory conversion should be allowed on unallocated/allocated states only

Allocator wrapper logic -
  if folio is NOT owned by wrapper and inode is alive
    notify inode about folio put callback
guest_memfd logic -
  on notification from wrapper
    for last folio in hugepage, merge the folio

Allocator wrapper logic-
  if folio is NOT owned by wrapper and inode is alive
    notify inode about folio put callback
  if folio is owned
    for last folio in hugepage, merge the folio and release
guest_memfd logic-
  on notification from wrapper, release the folio to wrapper

Guest_memfd Usecases

- Initial faultable unallocated -
  - Non-CoCo VMs, SNP VMs
- Initial unfaultable unallocated -
  - TDX VMs
- Faultable allocated -> truncate pending
  - Memory ballooning with non-coco VMs
- Unfaultable allocated -> truncate ready
  - Inode cleanup
- Truncate ready -> faultable unallocated
  - Memory ballooning
- Truncate ready -> unfaultable unallocated
  - No real usecase yet



core-mm
base allocator
wrapper
guest_memfd

Initialization

CreateWrapperInstance(inode)

1) Create wrapper object
2) associate supplied inode with the wrapper object

Cleanup

ReleaseWrapperInstance(handle, inode)

1) Disassociate inode from the wrapper instance
2) Drop the refcount on wrapper instance

Runtime

TransferFolioOwnership(handle, folio)

Used by gmem in case of truncate operation

EnableFolioPutCallback(handle, folio)

Used by gmem in case of shared -> private callback

GetWrapperRef(handle)/PutWrapperRef(handle)

Used by gmem in case of split/merge folio ops.

SplitFolio(handle, folio)/MergeFolio(handle, folio)

AllocateFolio(handle)

core-mm->Wrapper

folio_put callback

inode is associated with wrapper object and folio is not owned by the wrapper

folio_put relay

- update offset states i.e. tready -> unallocated or
tpending to tready or ufapending -> ufa
- TransferOwnership to wrapper for tpending -> tready transition
- merge the folio when last folio does ufapending -> ufa transition.
 - Drop the refcount of wrapper object.

folio is owned by the wrapper

for last folio of hugepage
- Merge the folio
- free the folio back to base allocator
- drop the refcount of wrapper object

core-mm
base allocator
wrapper
guest_memfd