Contribution ID: **116** Type: **not specified**

Optimizing guest_memfd shared/private conversions

HugeTLB support in guest_memfd is making steady progress, and has also led to some new problems that come with huge page support. HugeTLB support currently relies on runtime folio restructuring (split/merge) for accurate refcount tracking that integrates well with other users of struct folio.

Current support ends up introducing significant cost in terms of conversion performance. Folio restructuring also means extra memory has to be kept around to support (temporarily) undoing HugeTLB vmemmap optimization.

I would like to get the community's opinions on the possible optimizations:

- Would it be possible for the guest to cooperate to improve conversion performance?
- Can we remove page structs for private memory altogether?

Primary author: TNG, Ackerley

Presenter: TNG, Ackerley

Session Classification: Confidential Computing MC

Track Classification: Confidential Computing MC