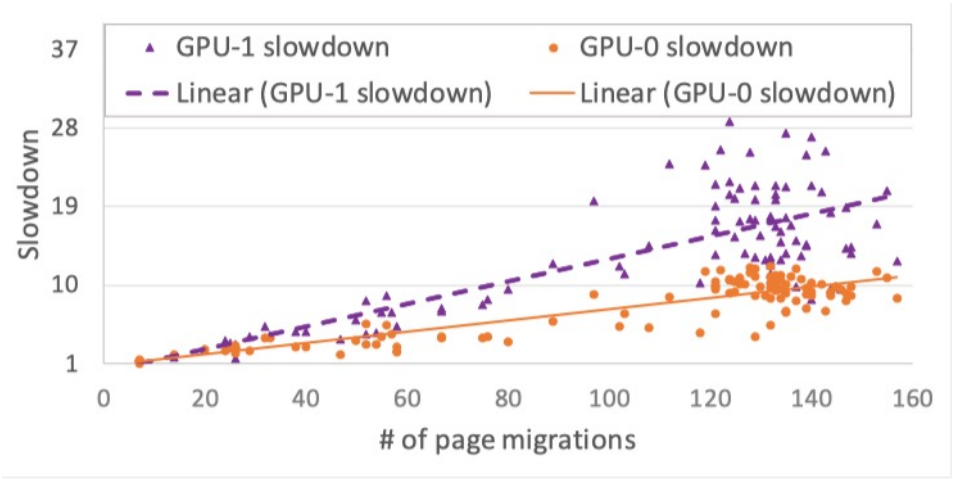
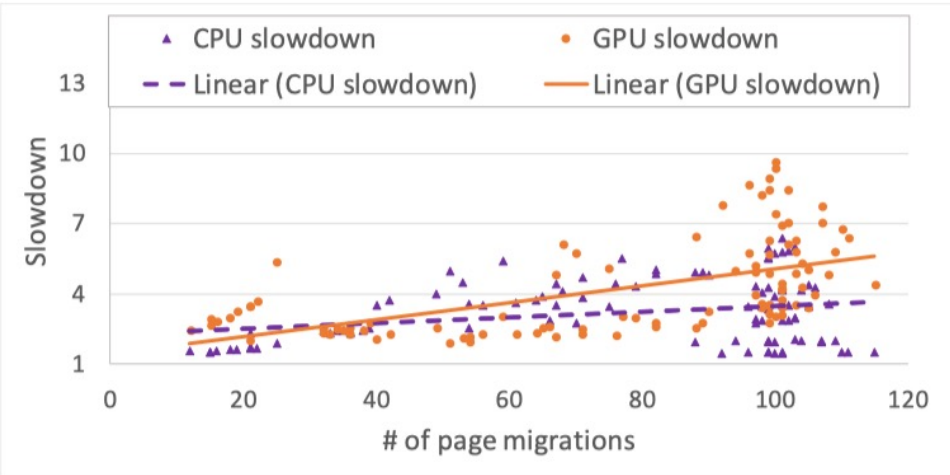
GAIA: An OS Page Cache for Heterogeneous Systems

Contributions

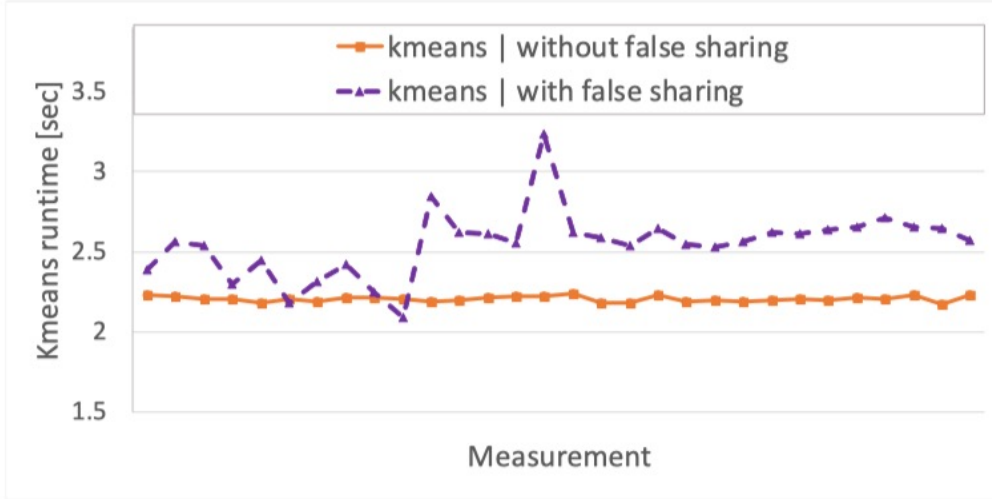
- Propose a unified page cache which eliminates false sharing by using a lazy release consistency (LRC) model
- Extend OS page cache to control the unified page cache and its consistency, without requiring CPU-GPU hardware cache coherence.
- Introduce a peer-caching mechanism and integrate it with the OS readahead prefetcher
  - enabling any processor accessing files to retrieve them from the best location



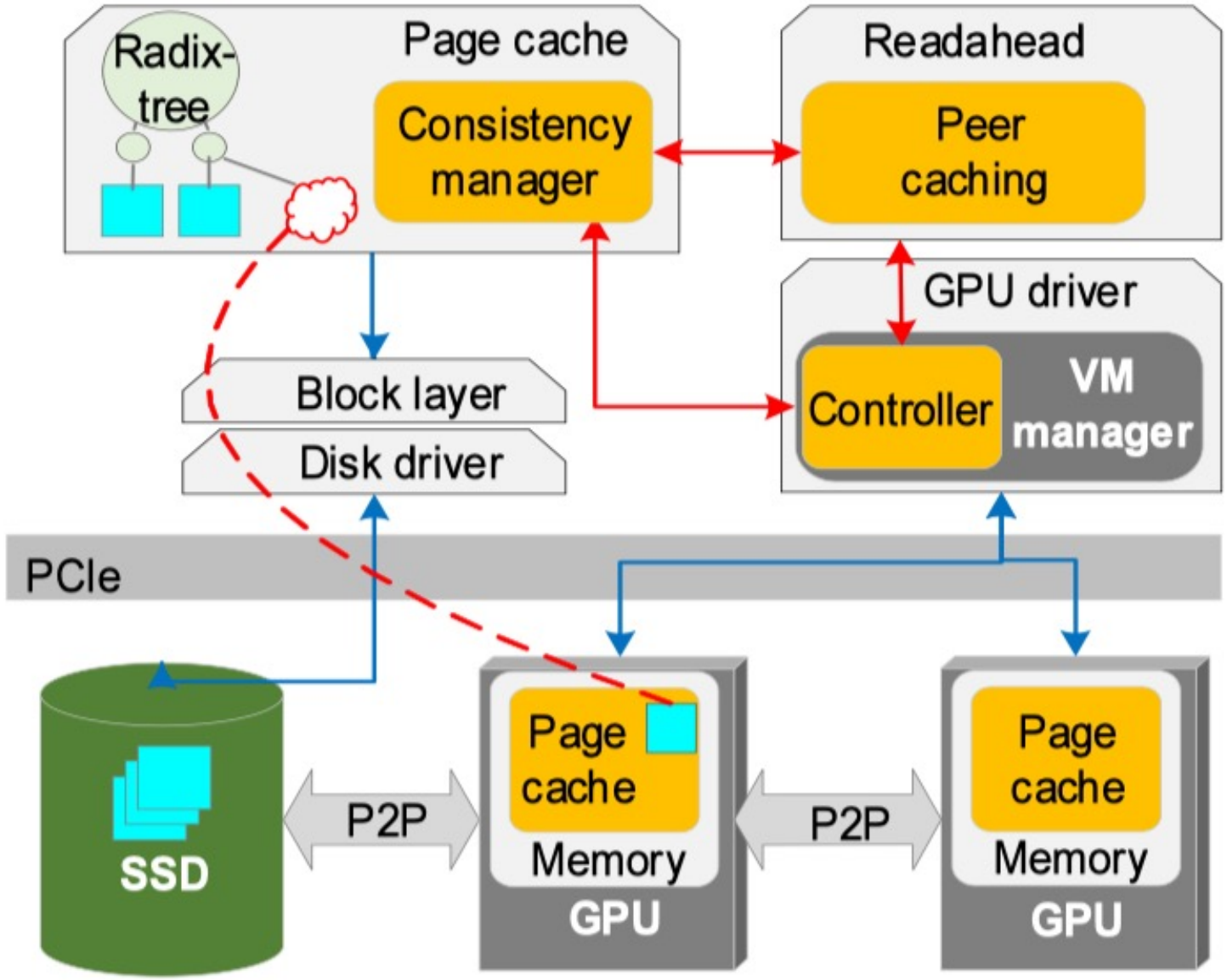
(a) False sharing between two GPUs



(b) False sharing between CPU and GPU

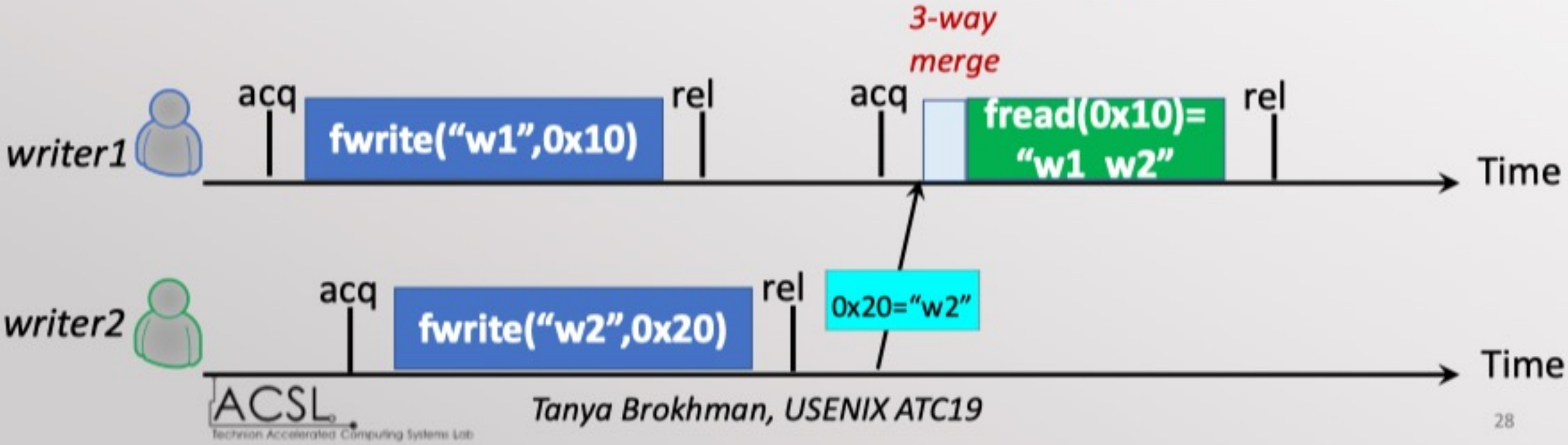


(c) The effect of false sharing in GPUs on an isolated CPU-only kmeans benchmark [33]


Radix-tree
Page cache
Consistency manager
Readahead
Peer caching
GPU driver
Block layer
Disk driver
Controller
VM manager
PCIe
SSD
P2P
Page cache
Memory
GPU
P2P
Page cache
Memory
GPU

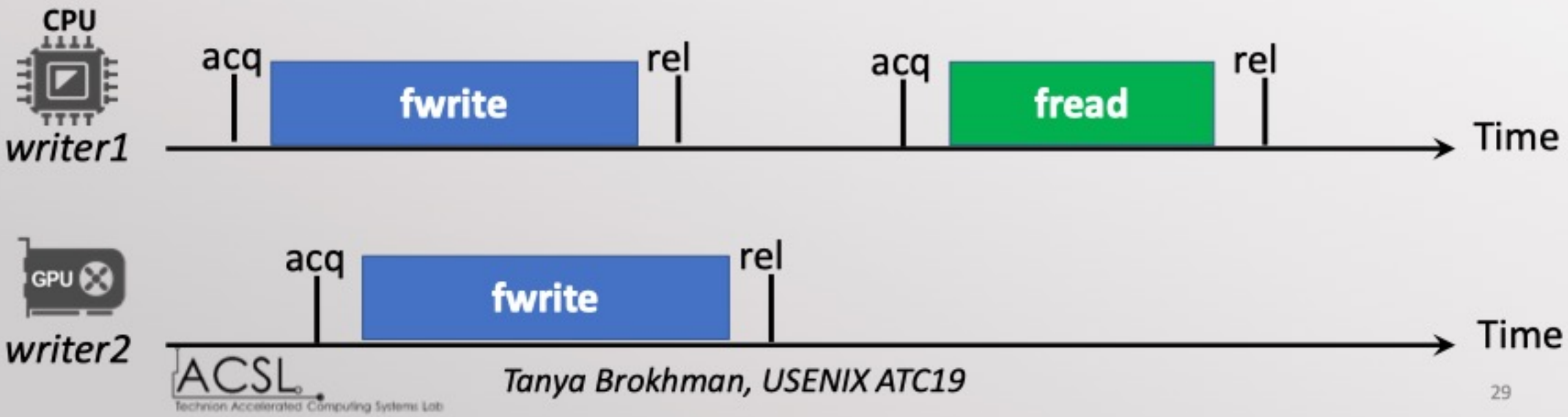
Lazy Release Consistency (LRC)

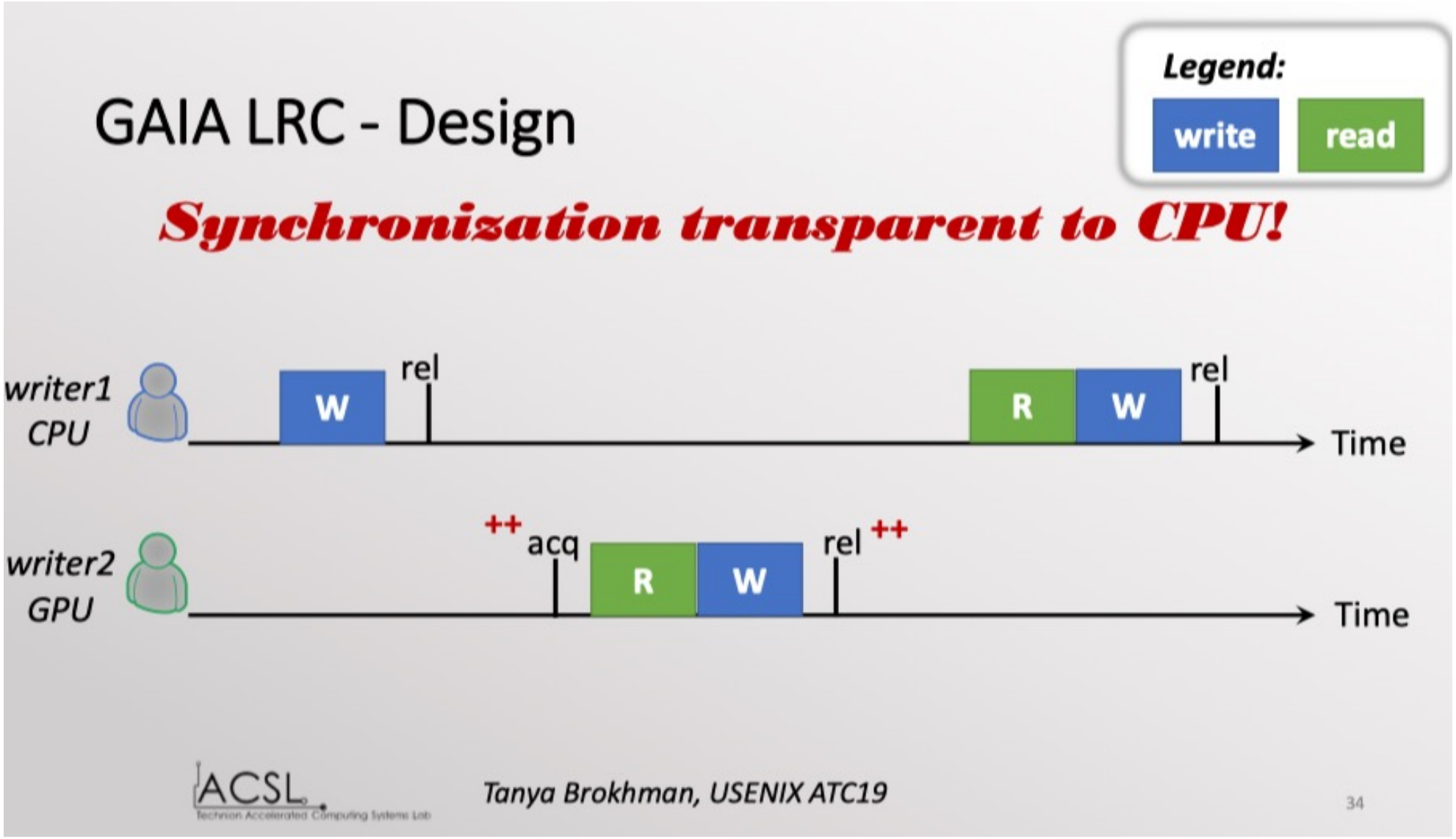
- Visibility of writes defined by *acquire* and *release* synchronization operations
  - The writes are visible after the writer *release*-s and the reader *acquire*-s
- The propagation of the updates is delayed until *acquire*



GAIA LRC

- Visibility of writes defined by *acquire* and *release* synchronization operations
  - The writes are visible after the writer *release*-s and the reader *acquire*-s
- The propagation of the updates is delayed until *acquire*





- All the synchronization disk access the copy to the between memories is done transparently for user by the operating system

GAIA - Conclusion

- Scalable weakly consistent page cache abstraction extended to GPU memory
  - Demonstrating the benefit of LRC for write-shared workloads
  - Support mapping large files into GPU address space, enabling on-demand I/O
- Backward compatible with legacy CPU and *unmodified* GPU kernels
  - Transparent consistency support for legacy CPU applications
- IO optimizations for legacy CPU applications

http://github.com/acsl-technion/gaia