Motivation

Hardware:

Software:
- Variable Workload
  - Write heavy
  - Highly Skewed
  - Large Size [KB]
  - Diurnal Pattern
- Concurrency bugs in popular open source

- Can we help developers to cover both situations?

Challenges
1. Semantic Preserving
2. Workload Patterns
3. Heterogeneous Latency

Design

Exploring Direction
Currently focusing on 2 first as a base => A scalable transactional synchronization on RDMA
1. What is the performance of existing synchronizations if there objects can be in Far Memory using existing approaches (w.r.t r/w ratio, skewness, object size)
2. Prototype design choices: - - synchronization overhead, + + local object access
   a. Object Placement: temporal objects, non temporal objects
   b. Elastic Log Space: on-demand log space allocation
   c. GC Design: Opportunistic GC, Pessimistic GC; (De)couple GC
   d. Cooperative Scheduling: Latency hiding