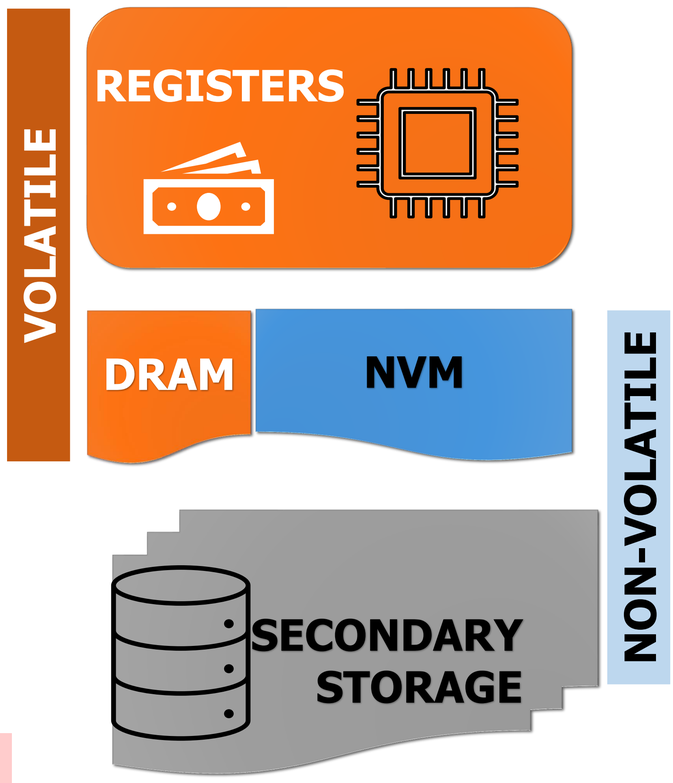
Highly-Efficient Persistent FIFO Queues

Panagiota Fatourou, Nikos Giachoudis, George Mallis

Motivation, Challenges, Contributions

Persistence Principles for Efficiency [Fatourou, et al., PPoPP 2022]

1. **Maintain the number of persistence instructions low**
2. **Persist those shared variable that will cause small persistence cost**
   - ❖ **avoid persisting highly-contented shared variables**
3. **Persist consecutive data**
   - ❖ **pwbs are applied on cache-line granularity**

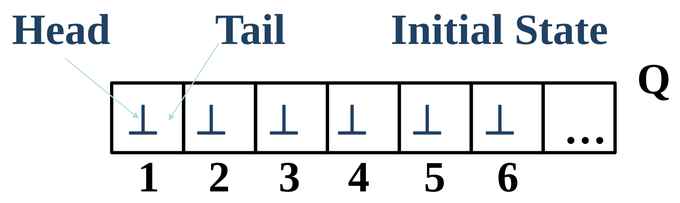


Expensive persistence instructions
- **Pwb:** asynchronous write back of a var
- **Psync:** block until write back has taken effect

Goal: **achieve low** persistence overhead when designing data structures

Contributions

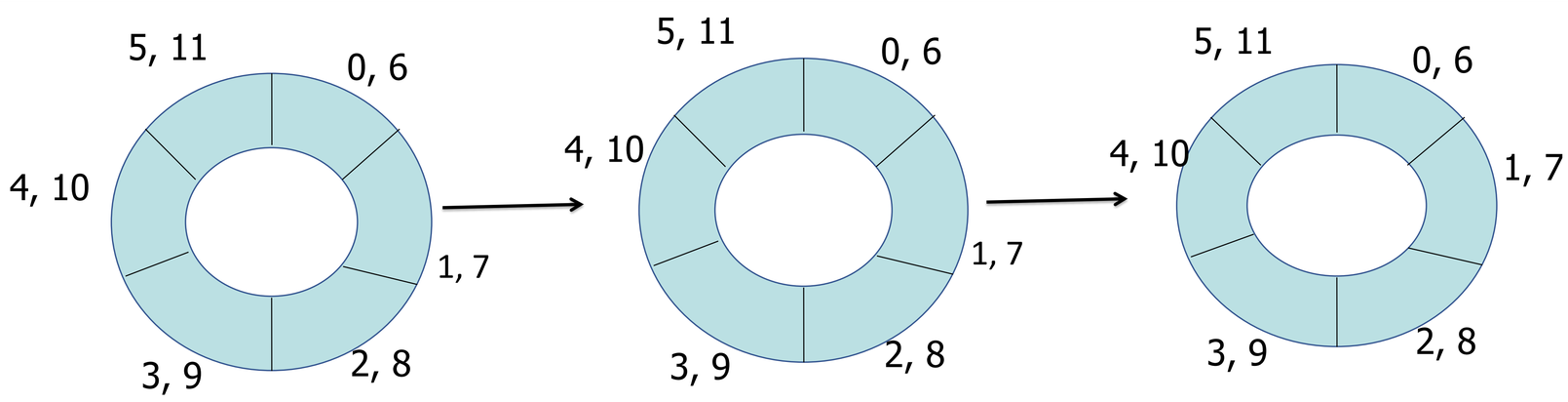
- PerLCRQ, a persistent implementation of FIFO queue: significantly outperforms previous implems
  - ❖ It illustrates how to efficiently persist algorithms that use Fetch&Add.
  - ❖ Promotes the idea of supporting persistence on top of SotA algs for the conventional setting
- PerLCRQ introduces techniques for reducing the persistence cost that could be of general interest.
- **Framework to simulate failures and measure the recovery cost of algs.**



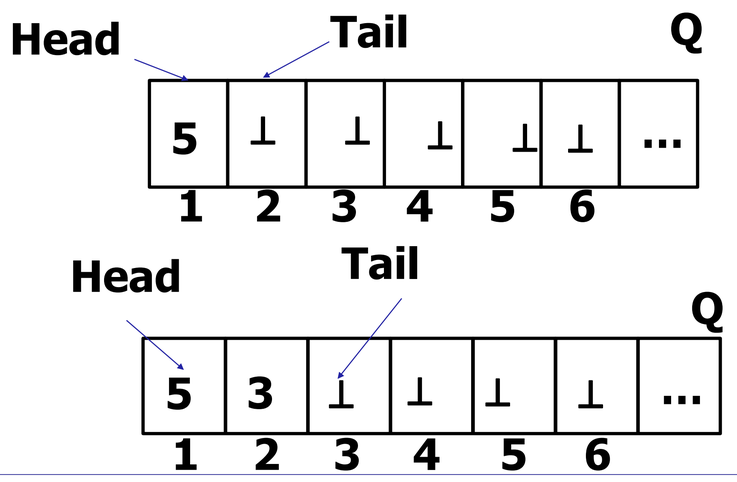
Persistent LCRQ

Basic Idea: Enqueue

1. Perform FAI on Tail to get next available position (pos) of Q
2. Perform Get&Set on Q[pos] to store the new value there
3. If result is ⊥
   PERSIST Q[pos]
   return
4. Otherwise, repeat steps above

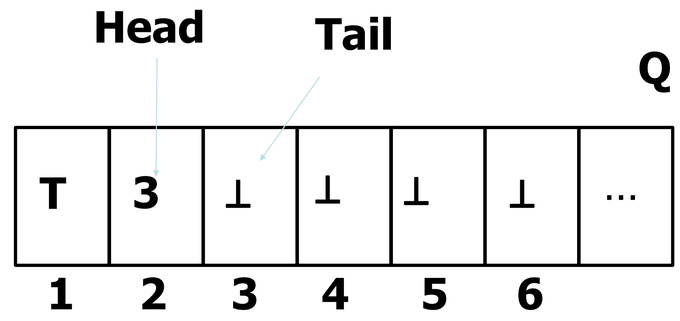


LCRQ employs a list of circular queues.

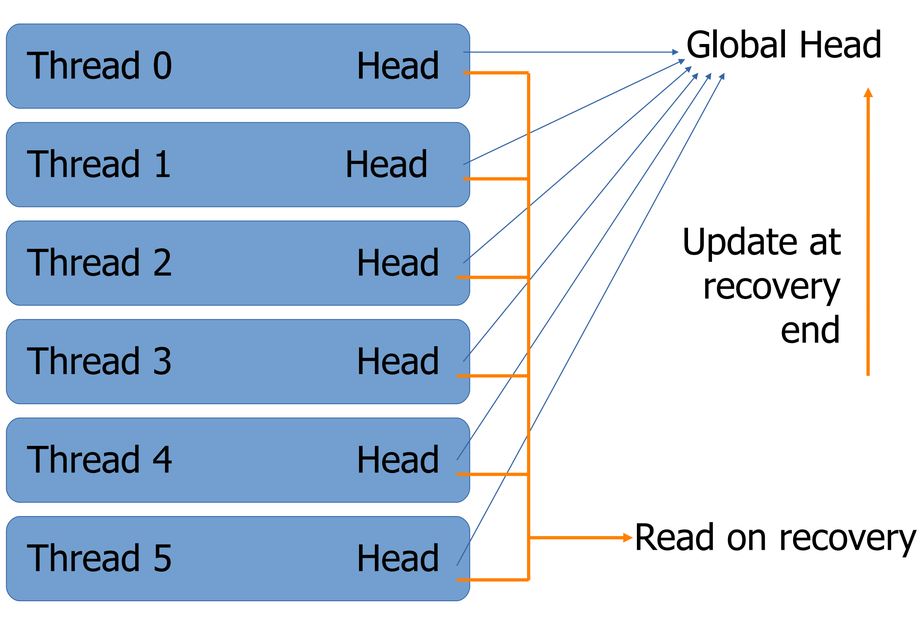


Basic Idea: Dequeue

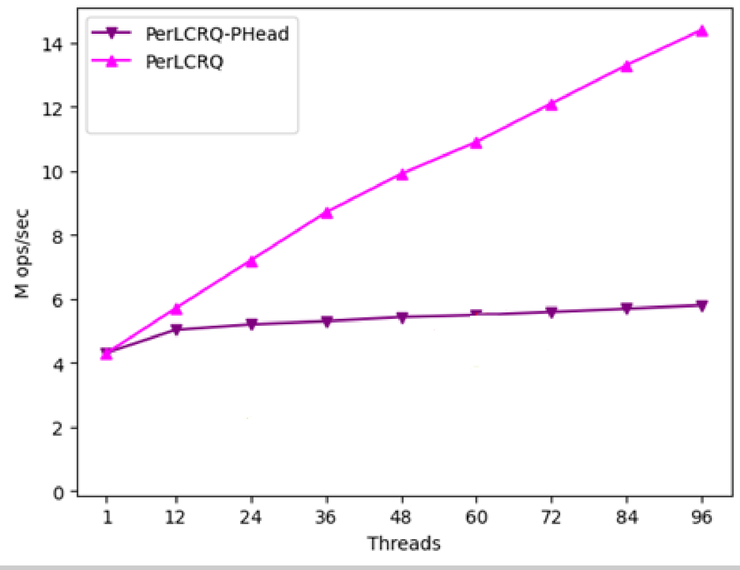
1. Perform FAI on Head to get next position of Q to dequeue from
2. Perform Get&Set on Q[pos] to store T there
3. If result is not equal to ⊥
   - PERSIST Q[pos]; return result
4. If Head > Tail
   - PERSIST Q[pos] ; return EMPTY
5. Otherwise, repeat above steps

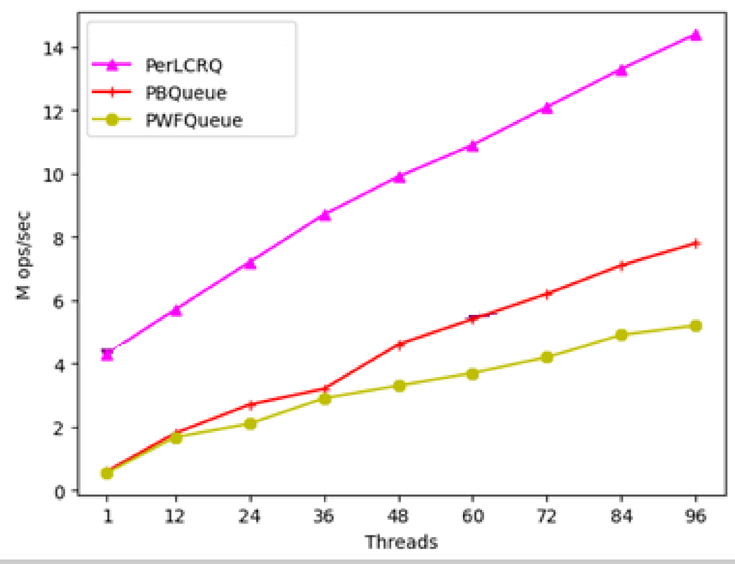


Technique for Reducing Persistence Cost



Evaluation





Machine with 2 Intel(R) Xeon(R) 5318 processors with 24 cores each, equipped with 128 GB Intel Optane 200 Series Persistent Memory.

INSTITUTE OF COMPUTER SCIENCE
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
UNIVERSITY OF CRETE
H.F.R.I. Hellenic Foundation for Research & Innovation
persist

This project has received funding from the Hellenic Foundation for Research and Innovation under the 2nd Call for H.F.R.I.'s Research Projects to Support Faculty Members & Researchers, Agreement No 03684.

Contact
George Mallis
ICS-FORTH
gmallis@ics.forth.gr