

Lock-free Linearizable 1-Dimensional Range Queries

Bapi Chatterjee

Distributed Computing and Systems Group
Department of Computer Science and Engineering



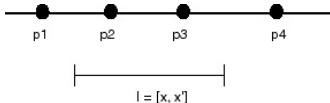
Chalmers University of Technology, Sweden

8th WTTM 2016
July 25, 2016

Introduction

The problem statement

- Given: Set of data-points on the real line and an interval thereon.



- Setup: Dynamic and Concurrent addition and removal of data-points.
- Output: The data-points in the interval.
- Requirement:
 - Lock-freedom.
 - Linearizability.

Problem relevance

- Ubiquitous multi-core systems.
- The ordered Set ADT provides $\text{size}(T\ a, T\ b)$ that is implemented in sequential setup.
- Increasing availability of efficient lock-free ordered sets.
- Linearizability provides an aligned view with respect to a real-time line.
- Growing popularity of In-memory Databases. Some with lock-free index structures e.g. MemSQL uses lock-free skip-lists.

Existing linearizable concurrent range search implementations

- ① Brown and Avni [OPODIS '12]: Lock-free k-ary search tree based; Lock-free; Double-scan and validation.
- ② Avni et al. [PODC '13]: STM based approach on a variant of Skip-list with “*fat*” nodes; Blocking.
- ③ Sagonas and Winblad [LCPC Workshop '15]: Locking tree structure based; Blocking.

Presented Method

Motivation for an alternate approach

- Progress guarantee.
- Independence from the performance of a particular STM.
- Avoiding multi-scan and restart.

Overview

Afek et al. '93: Snapshot



Attiya et al. '08, Imbs et al. '09: Partial Snapshot



Brown and Avni '12: kst based Range Search

Jayanti '05: Snapshot

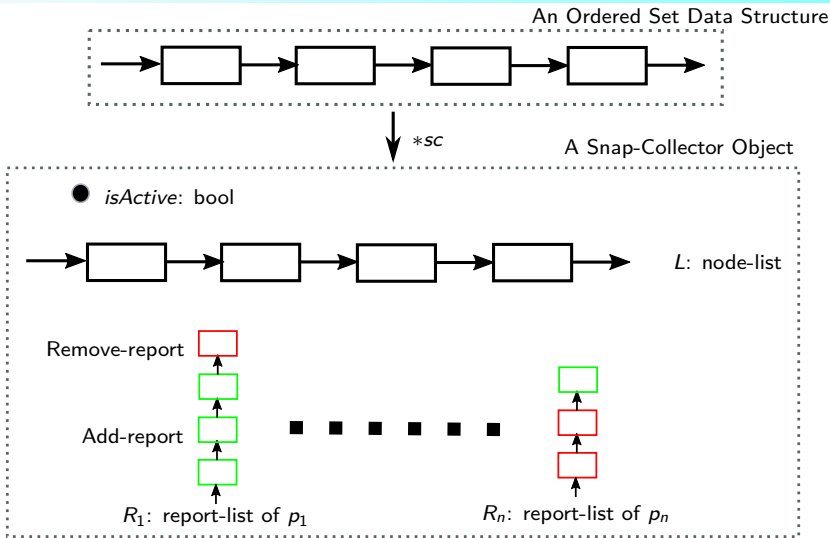


Petrank et al. '13: Lock-free Iterators

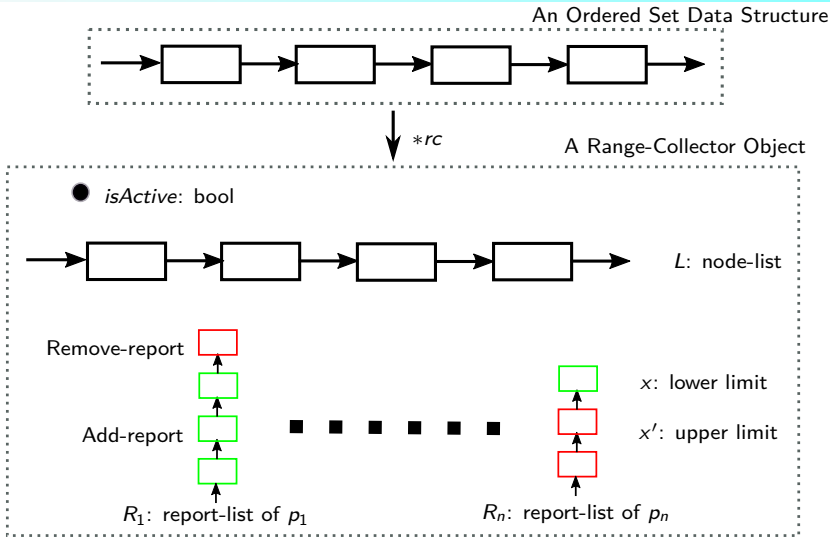


Our method

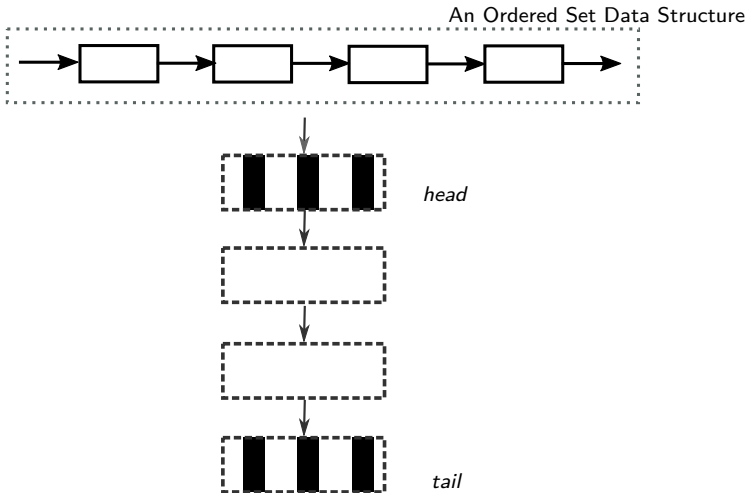
Snap-collector



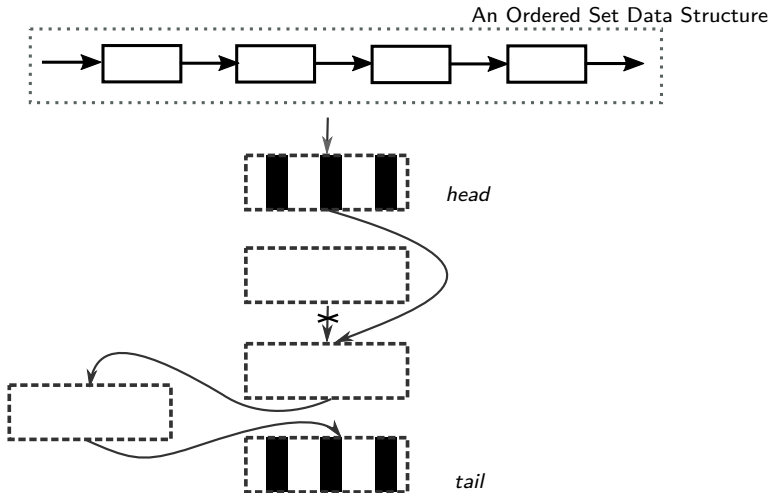
Range-collector



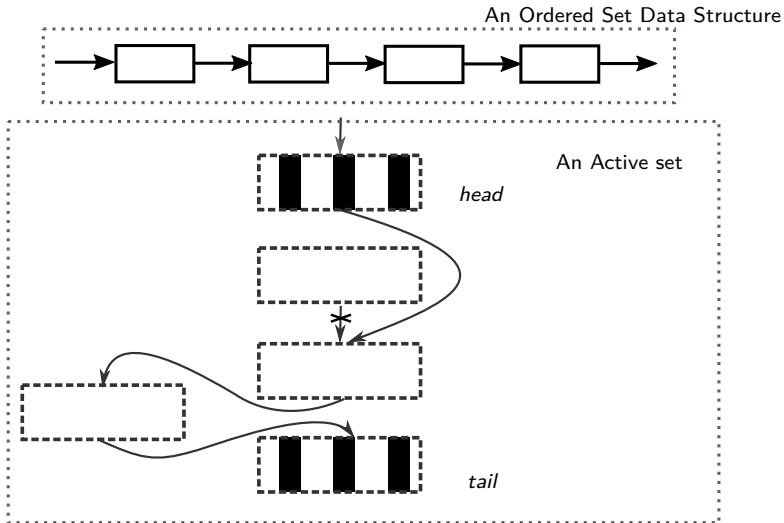
Lock-free data structure with Range-collector-list



Lock-free data structure with Range-collector-list



Lock-free data structure with Range-collector-list



Performance

Performance

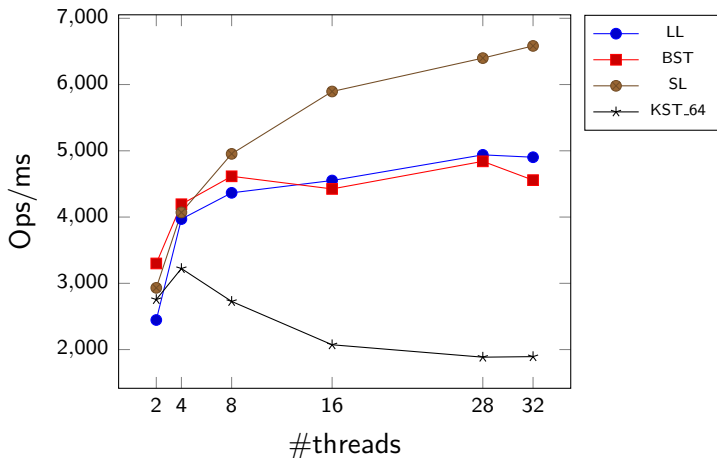


Figure: 5% ADD, 5% REM, 89%CON, 1% RSEARCH, $0 \leq x < 10^2$

Performance

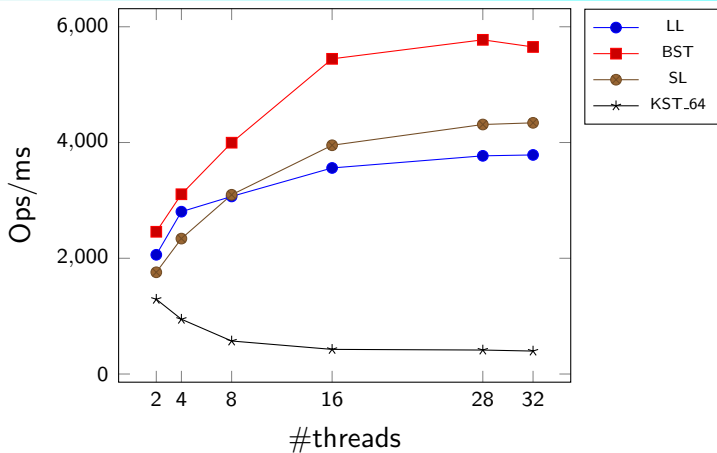


Figure: 25% ADD, 25% REM, 49% CON, 1% RSEARCH, $0 \leq x < 10^2$

Performance

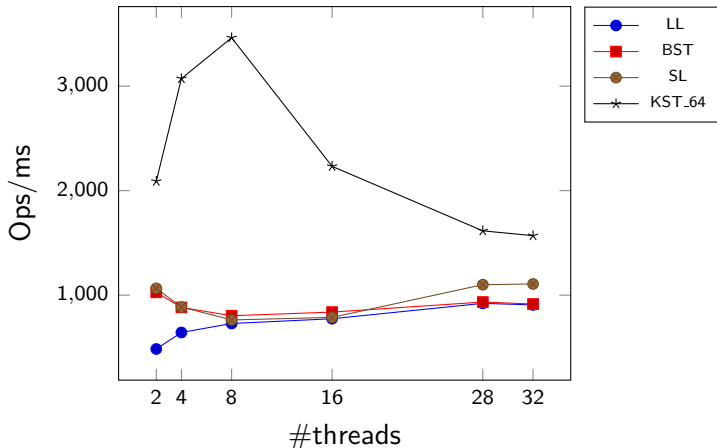


Figure: 5% ADD, 5% REM, 89%CON, 1% RSEARCH, $0 \leq x < 10^3$

Performance

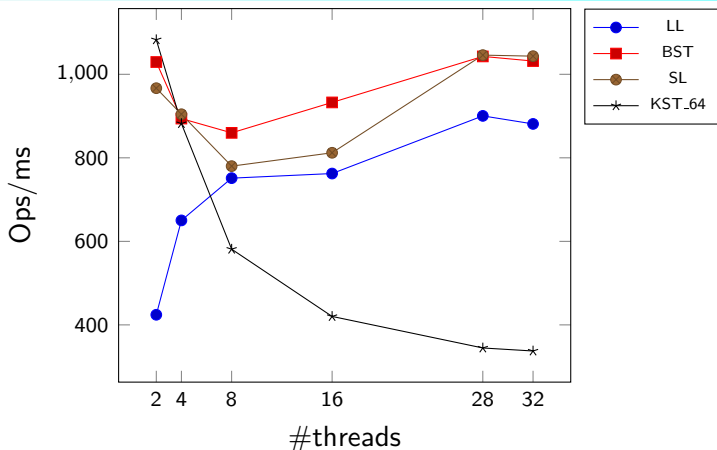


Figure: 25% ADD, 25% REM, 49%CON, 1% RSEARCH, $0 \leq x < 10^3$

Summary

Summary

Summary

- A generic linearizable lock-free 1-dimensional range search.
- No multiple-scan, no invalidation.
- Scalability does not deteriorate with high number of updates.