accu 2023

INTRODUCTION TO EPOCH-BASED MEMORY RECLAMATION

JEFFREY MENDELSOHN

Introduction to Epoch-Based Memory Reclamation

Bloomberg

Engineering

What to do When no Thread is Watching

ACCU 2023
April 21, 2023

Jeffrey I. Mendelsohn
Software Infrastructure Team Leader, BDE

TechAtBloomberg.com

Welcome

- Hi! I’m Jeff!
- Please “interrupt” with questions, prefer to be interactive.
- Goal is to introduce and promote epoch-based algorithms.

TechAtBloomberg.com

Bloomberg

Engineering

Agenda

- Terms
- Memory Reclamation is Hard
- Hazard Pointers Memory Reclamation
- Epoch-Based Memory Reclamation
- Qualitative Comparison
- The Real Value

TechAtBloomberg.com

Bloomberg

Engineering

Terms

- Memory Reclamation
- Epoch-based Algorithm
- Quiescent State Algorithm (for comparison, this slide only)
- Hazard Pointers

TechAtBloomberg.com

Bloomberg

Engineering

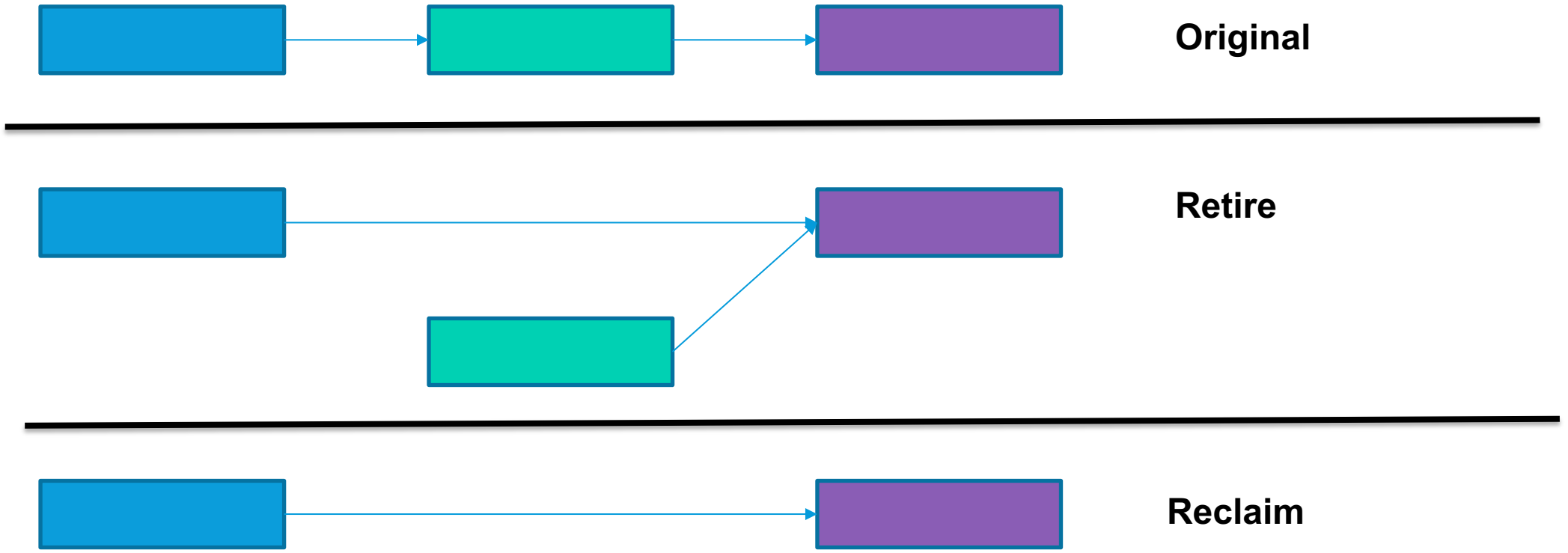
Memory Reclamation is Hard (1/2)

- Multiple threads may be reading an object.
- Accessing a deleted object is undefined behavior.
- Break the operation in a “retire” and a “reclaim” step.

Bloomberg

Engineering

Memory Reclamation is Hard (2/2)



TechAtBloomberg.com

Bloomberg

Engineering

Hazard Pointers Memory Reclamation

- Collection of object addresses that should not be reclaimed.
- May be a data structure specific collection.
- While accessing an object, address is added to the collection.
- Set of retired objects is maintained.
- Periodically reclaim retired objects not in protected collection

Epoch-Based Memory Reclamation (1/2)

- Classically, an epoch and three sets of retired objects.
- Track threads in waiting-to-reclaim, current-ish, and current.
- New threads join current.
- Reclaim from waiting-to-reclaim when no threads associated.
- After reclamation, update epoch.
- Retired objects are being added to current-ish and current.

Epoch-Based Memory Reclamation (2/2)

Practical Lock Freedom, Keir Frasier:

https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-579.pdf

Comparative Performance of Memory Reclamation Strategies for Lock-free and Concurrently-readable Data Structures, Thomas Hart:

http://www.cs.toronto.edu/~tomhart/papers/tomhart_thesis.pdf

Qualitative Comparison

- Traversing a list.
- Traversing a balanced tree.
- More threads than cores.
- Ease of implementation.

TechAtBloomberg.com

Bloomberg

Engineering

The Real Value

- **Alternative to hazard pointers for memory reclamation.**
- **Epoch-based concept is extensible to other domains.**

Thank you!

https://www.TechAtBloomberg.com/cplusplus

Check out our open roles
https://www.bloomberg.com/careers

Bloomberg

Engineering

TechAtBloomberg.com