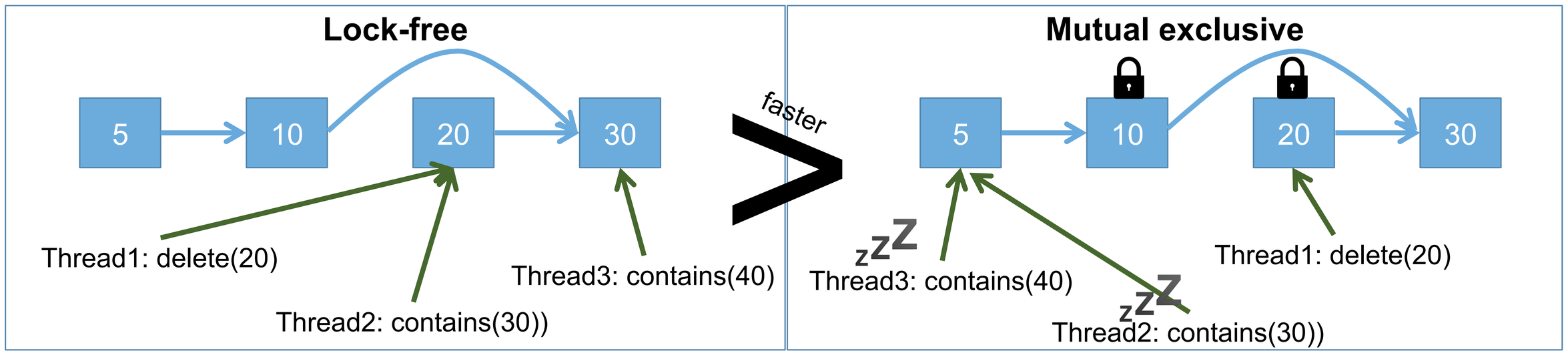
Memory Management for Lock-Free Data Structures with Optimistic Access

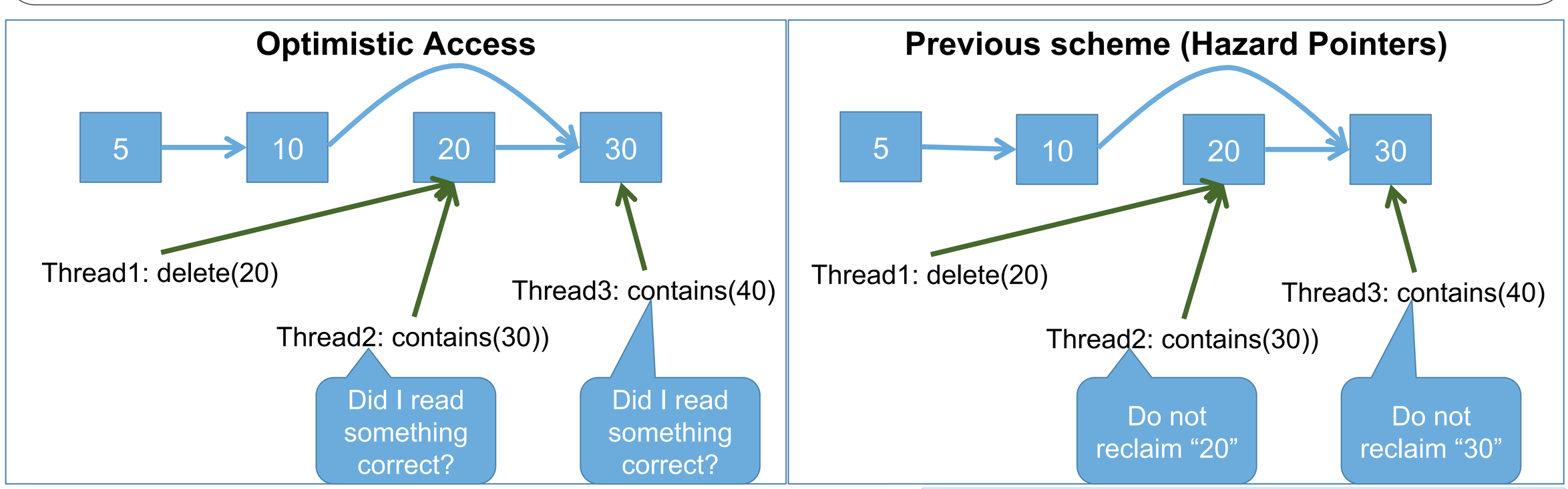
Nachshon Cohen and Erez Petrank

Background: multi-core system are popular. Efficient concurrent data structures are highly important.
Techniques: lock-free vs. mutual exclusive.



Problem: recycle deleted nodes? Hard to know who is referencing the deleted node.
Complex problem for lock-free data structures!

Previous paradigm: after reclamation, must not access memory
Optimistic Access: after reclamation, can access memory. Just avoid using the stale data.



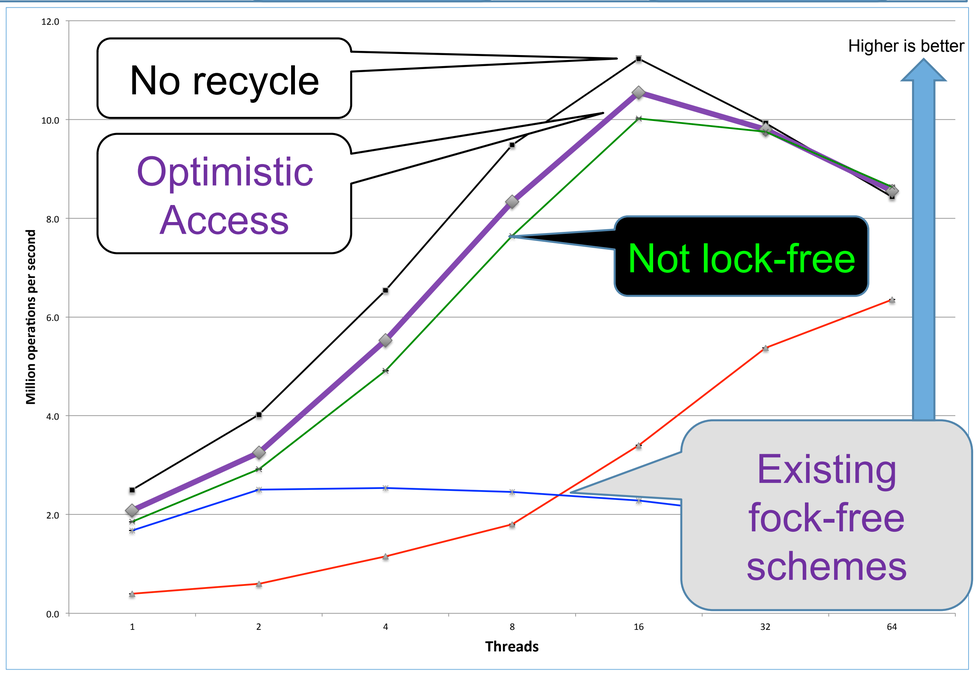
Optimistic Access: on every access check a flag for read validity.
If read stale data: restart the operation – traverse the list again.
Benefits: significantly faster than writing to shared memory!
Restarts are very rare, do not affect performance.

Some more:

Guarantee progress, even when threads fail/delay.

Automatic:

- **Applicable to many lock-free data structures.**
- **Easy to apply.**



Presented in SPAA’15 and OOPSLA’15