

Multi-granularity Locking on Multi-tier Data Structures

Bing Yang, Kenneth B. Kent, Eric Aubanel
University of New Brunswick, Faculty of Computer Science
Tobi Agila
IBM Canada
{bing.yang, ken, aubanel}@unb.ca, atobia,@ca.ibm.com

INTRODUCTION

Multi-granularity locking is a concept originating from Database Management Systems. The lock granularity defines the size of data items being guarded by a lock for synchronization. A fine-grained lock allows more concurrency, but increases the frequency of lock requests and releases. A coarse-grained lock introduces less lock overhead, but limits the degree of concurrency. Multi-granularity locking is necessary for synchronization on multi-tier data structures (like nested C structs) which organize data hierarchically and allow data shared among different tiers. Multi-granularity locking helps to achieve a balance between locking overhead and concurrency but also brings new challenges for synchronization. Our research studies the packed object data model introduced in IBM's J9 VM and proposed solutions for multi-tier data synchronization.

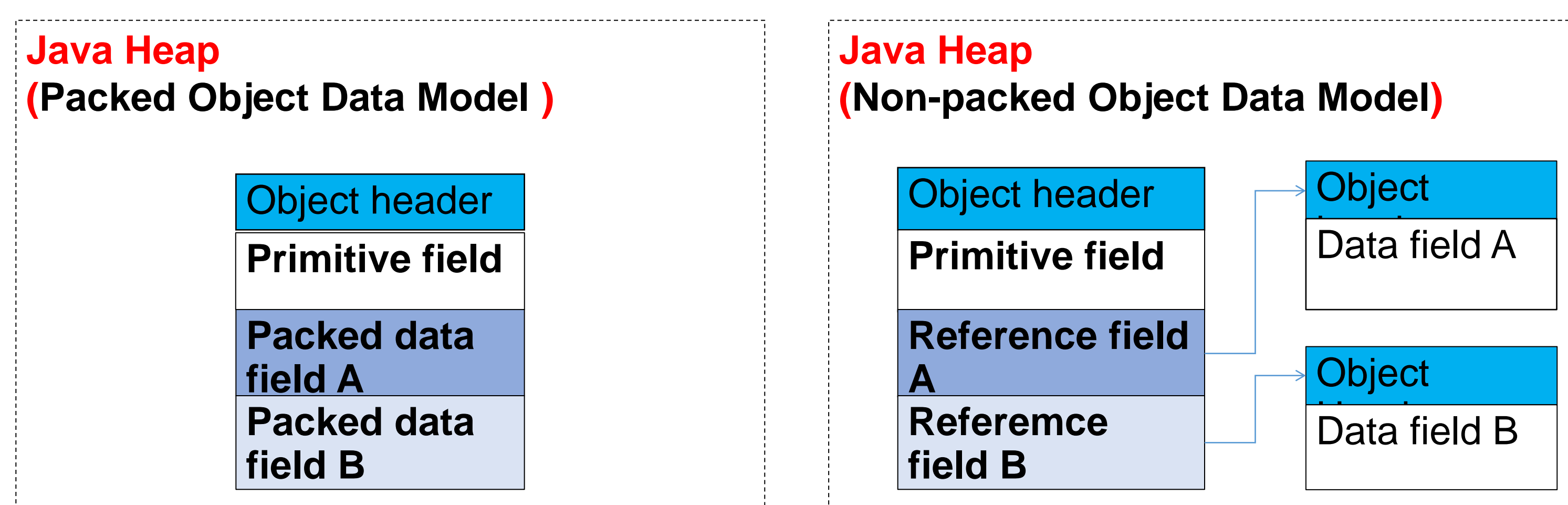


Figure 1: Packed object data model and traditional Java object data model

Problem Statement

- Multiple threads could arbitrarily get access to any level of the nested PackedObject.
- Detecting the overlapping data from different tiers and maintaining the monitor association with the PackedObjects are the main challenges.

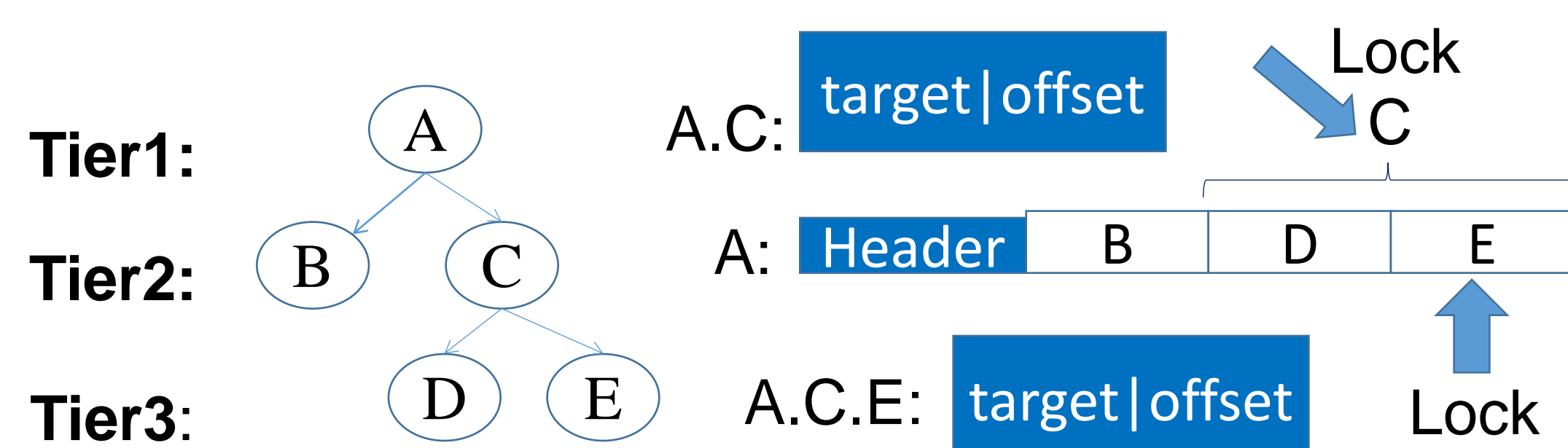


Figure 2: Memory layout of a three-tier PackedObject

```
@Packed class OneTierClass extends PackedObject{
    int value;
}
@Packed class TwoTierClass extends PackedObject{
    OneTierClass D;
    OneTierClass E;
}
@Packed class ThreeTierClass extends PackedObject{
    OneTierClass B;
    TwoTierClass C;
}
ThreeTierClass A =
PackedObject.newPackedObject(ThreeTierClass.class);
```

Technical Approaches

PackedSync: A monitor-based approach without relying on lock-word, only works with two tier packedObjects.

MultitierPackedSync: A optimized concurrent linked-list algorithm (lazy-list), works with multiple tier PackedObjects.

- Postpone physical deletion of node.
- Reduce overhead of unnecessary repetition of insertion and deletion.

Concurrent Multiway tree (Left-Child Right-Sibling representation tree)

- Tree structures are better in representing the containing or ordering relationship among the PackedObjects in different tiers.
- Average traversal distance for the target node in tree structures is shorter than in linked lists.
- Concurrent Search, Insert and Delete Methods allow multiple threads to concurrently manipulate the multiway tree structures.

Multi-granularity locking scheme

- Based on the concurrent multiway tree
- Allows multiple threads to concurrently update different tiers of PackedObjects and maintain consistency.
- Provides a more efficient approach to synchronization on multi-tier data structures.

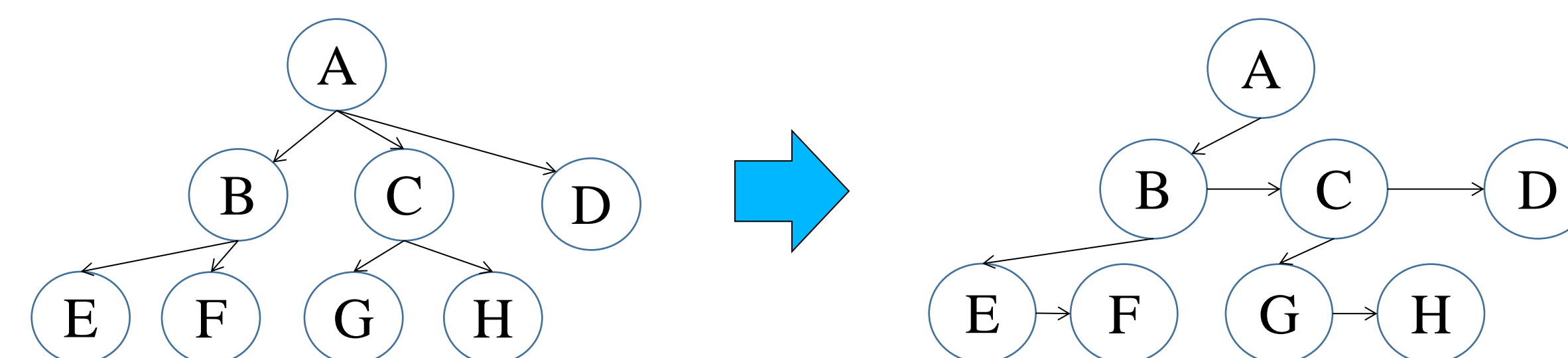


Figure 3: A multiway tree represented in a Left-Child Right-sibling tree