



UPPSALA
UNIVERSITET

UPTEC IT 24020

Degree project 30 credits

June 2024

Addressing Fragmentation in ZGC through Custom Allocators

Leveraging a Lean, Mean, Free-List Machine

Joel Sikström



UPPSALA
UNIVERSITET

Abstract

The Java programming language manages memory automatically through the use of a garbage collector (GC). The Java Virtual Machine provides several GCs tuned for different usage scenarios. One such GC is ZGC. Both ZGC and other GCs utilize bump-pointer allocation, which allocates objects compactly but leads to the creation of unusable memory gaps over time, known as fragmentation. ZGC handles fragmentation through relocation, a process which is costly. This thesis proposes an alternative memory allocation method leveraging free-lists to reduce the need for relocation to manage fragmentation.

We design and develop a new allocator tailored for ZGC, based on the TLSF allocator by Masmano et al. Previous research on the customization of allocators shows varying results and does not fully investigate usage in complex environments like a GC.

Opportunities for enhancements in performance and memory efficiency are identified and implemented through the exploration of ZGC's operational boundaries. The most significant adaptation is the introduction of a 0-byte header, which leverages information within ZGC to significantly reduce internal fragmentation of the allocator. We evaluate the performance of our adapted allocator and compare it to a reference implementation of TLSF. Results show that the adapted allocator performs on par with the reference implementation for single allocations but is slightly slower for single frees and when applying allocation patterns from real-world programs. The findings of this work suggest that customizing allocators for garbage collection is worth considering and may be useful for future integration.

Faculty of Science and Technology

Uppsala University, Uppsala

Supervisor: Erik Österlund Subject reader: Tobias Wrigstad

Examiner: Roland Bol