

lgmalloc: Predictive Memory Allocator

Luca Martijn Goddijn
luca.goddijn@gmail.com
Independent Researcher
Amsterdam, Netherlands

Abstract

Memory allocation represents a critical bottleneck in multi-threaded applications. Traditional allocators like `tcmalloc` suffer from central contention points that limit scalability, while thread-local implementations require substantial per-thread memory overhead. `lgmalloc` shifts the demand-reactive paradigm that standard allocation algorithms use, to a preemptive and predictive heuristic approach. The approach leverages the fundamental observation that call sites demonstrate predictable allocation patterns. By parsing the executable’s ELF format before the program’s entry point, `lgmalloc` performs static analysis of call sites to reveal allocation patterns and generate confidence scores for size classes. Based on the confidence scores, the system utilizes optimized code paths to the backend allocation API, reducing allocation latency. When confidence scores are low, the system falls back to reliable, standard allocation strategies. The thread-local design eliminates lock contention while substantially reducing per-thread memory overhead by optimizing the pre-allocated size classes each thread requires.

`lgmalloc` is available at: <https://github.com/Arty3/lgmalloc>

Keywords

memory allocation, static analysis, predictive optimization, thread-local storage, heuristics, heap

- 1 Introduction
- 2 Background and Related Work
- 3 System Design
- 4 Implementation
- 5 Evaluation
- 6 Discussion
- 7 Conclusion
- References

Permission to make digital or hard copies of all or part of this work for personal or internal use, or the internal or personal use of specific clients, is granted by ACM for libraries registered with ACM, for non-profit organizations registered with ACM, and for non-profit individuals registered with ACM, on the basis that copies are made for non-profit use and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference '25, Washington, DC, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>