

---

# 语言虚拟机内存管理研究取得进展

作者：writer 来源：中国科学院

本文原地址：<https://www.iikx.com/news/progress/34544.html>

*本文仅供学习交流之用，版权归原作者所有，请勿用于商业用途！*

## 语言虚拟机内存管理研究取得进展

。近日，中国科学院软件研究所科研团队在语言虚拟机内存管理方面取得进展，涉及垃圾回收（GC）算法的跨语言性能评估以及面向大数据系统的新型垃圾回收器设计。

在现代编程语言中，GC负责自动管理内存分配和回收，其效率直接影响用户程序的性能。目前，学术界和工业界主要通过对比评估单一语言运行环境下的GC实现，来分析性能差异指导GC优化，缺少直接实现不同语言运行环境下的GC性能差异评估的方法。

针对这一不足，研究团队提出了GEAR方法，通过自动构建适用于不同语言运行环境的一致性GC工作负载，实现跨语言GC性能评估。GEAR的核心在于设计了一组与运行环境无关的内存操作原语（MOP），用于描述影响GC行为的关键内存使用信息。GEAR可以将这些MOP程序自动转换为目标语言的代码，并生成一致的GC测试负载。为保证测试负载的真实性，团队对Java虚拟机进行插桩，记录实际Java应用程序执行期间的内存操作，并将其转换为MOP程序，进而适配到其他语言运行环境。

实验表明，GEAR能够在Java、Go和C#三个广泛使用的运行环境生成一致的GC工作负载。基于此，团队对比了这三种语言的GC实现情况，揭示了6项性能差异，并分析了可能原因，为未来GC优化提供了参考依据。

相关成果被ICSE

2025会议录用。研究工作得到国家自然科学基金和中国科学院相关项目的支持。

当前，Apache Flink、Spark等主流大数据框架普遍运行于JVM平台，其内存管理依赖JVM虚拟机的GC算法。然而，大数据应用通常产生大量长生命周期的数据对象，这与传统GC算法针对短生命周期对象优化的设计假设存在差异，导致GC效率低下，影响系统性能。

针对这一问题，研究团队面向大数据系统，设计并提出了新型跨层次垃圾回收器BridgeGC，通过框架层与GC层的协同设计实现内存管理优化。具体来说，BridgeGC在框架源码层面采用极简注解方式标记关键数据的创建和释放点，帮助GC层识别数据对象及其生命周期；BridgeGC在GC层设计了分离式内存布局，将普通对象和大数据对象分区管理，提高内存访问效率；BridgeGC通过自适应GC回收策略，根据对象生命周期动态调整数据处理时期，避免冗余标记等不必要操作。

研究团队将BridgeGC集成至OpenJDK 17的ZGC回收器，并在Apache Flink、Spark等主流大数据系统进行验证。实验数据显示，BridgeGC在保持内存使用效率的同时将GC耗时降低了31%至82%。

---

此外，BridgeGC提升了端到端应用性能，且对业务系统透明，无需修改应用代码。

相关成果被ACM TACO录用。研究工作得到国家自然科学基金等的支持。

研究团队单位：软件研究所

更多 科学进展 请访问 <https://www.iikx.com/news/progress/>

本文版权归原作者所有，请勿用于商业用途，[爱科学iikx.com](http://www.iikx.com)转发