

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НИЗКОУРОВНЕВЫХ МЕТОДОВ ОПТИМИЗАЦИИ ВЫСОКОНАГРУЖЕННЫХ СИСТЕМ НА ЯЗЫКЕ GO

Алиев Т.Р.¹

Научный руководитель – доцент, канд. физ.-мат. наук Арефьев А.В.¹

¹ФГАОУ ВО «ГУАП»

timealive@mail.ru

Введение. Развитие облачных технологий и микросервисной архитектуры диктует высокие требования к производительности серверного программного обеспечения. Язык программирования Go позиционируется как эффективный инструмент для создания высоконагруженных систем с огромным количеством данных и серверов. Однако, для достижения максимальной пропускной способности и минимизации задержек не всегда достаточно применения только глобальных архитектурных решений и горизонтального масштабирования. Актуальной научной проблемой является систематизация и экспериментальная оценка влияния низкоуровневых механизмов оптимизации, таких как управление памятью, настройка рантайма и использование специфических инструкций CPU, на ключевые метрики производительности при пиковой нагрузке.

Основная часть. Цель работы заключается в экспериментальном определении вклада низкоуровневых методов оптимизации в производительность высоконагруженного сервиса на языке Go при обработке интенсивного потока структурированных данных. Для достижения цели разработан экспериментальный стенд, реализующий четыре идентичных по функциональности HTTP-эндпоинта, различающихся исключительно способом реализации. В качестве базовой линии используется реализация, опирающаяся на стандартные механизмы управления памятью, автоматическую сборку мусора и типичные паттерны обработки запросов [1]. Вторая версия ориентирована на минимизацию аллокаций и уменьшения частоты и длительности пауз сборщика мусора [2]. Третья реализация использует профиль-управляемую оптимизацию [3], позволяющую компилятору адаптировать стратегию инлайнинга и размещения кода на основе реального профиля исполнения. Четвёртая версия включает аппаратно-зависимую оптимизацию вычислительно-интенсивных участков за счёт векторизации и использования SIMD-инструкций.

В рамках тестирования фиксируются метрики количества аллокаций и объёма выделяемой памяти, загрузки CPU, характеристик работы сборщика мусора и задержек на хвосте распределения, что позволяет оценить влияние каждой техники на среднюю производительность и устойчивость при пиковых нагрузках.

Выводы. Проведенное исследование позволило дать количественную оценку эффективности низкоуровневых методов оптимизации в Go. Полученные результаты будут систематизированы в виде практических рекомендаций по выбору оптимального набора техник в зависимости от требований к задержкам и пропускной способности приложения. Материалы доклада могут быть использованы инженерами при проектировании и оптимизации критически важных компонентов высоконагруженных систем.

Список использованных источников.

1. Д. А. Хайруллин, К. Е. Чиркова, М. Р. Богданов ЭФФЕКТИВНЫЕ ПАТТЕРНЫ КОНКУРЕНТНОГО ПРОГРАММИРОВАНИЯ В GOLANG //

Международный журнал гуманитарных и естественных наук. 2025. №5-1 (104). URL: <https://cyberleninka.ru/article/n/effektivnye-patterny-konkurentnogo-programmirovaniya-v-golang> (дата обращения: 23.02.2026).

2. Денис Сергеевич Фадеев Исследование влияния параметров Golang Garbage Collector на производительность приложений и использование ресурсов системы // Вестник НГУ. Серия: Информационные технологии. 2025. №3. URL: <https://cyberleninka.ru/article/n/issledovanie-vliyaniya-parametrov-golang-garbage-collector-na-proizvoditelnost-prilozheniy-i-ispolzovanie-resursov-sistemy> (дата обращения: 23.02.2026).

3. Масандилов Илья ИНСТРУМЕНТЫ ПРОФИЛИРОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ GO-ПРИЛОЖЕНИЙ В РЕАЛЬНОМ ВРЕМЕНИ // Universum: технические науки. 2025. №9 (138). URL: <https://cyberleninka.ru/article/n/instrumenty-profilirovaniya-proizvoditelnosti-go-prilozheniy-v-realnom-vremeni> (дата обращения: 23.02.2026).