Programming Assignment: Оптимизация программы

You have not submitted. You must earn 1/1 points to pass.

Deadline Pass this assignment by January 20, 11:59 PM PST

Instructions My submission Discussions

hw3_bench.zip

Есть функция, которая что-то там ищет по файлу. Но делает она это не очень быстро. Надо её оптимизировать.

Задание на работу с профайлером pprof.

Цель задания - научиться работать с pprof, находить горячие места в коде, уметь строить профиль потребления cpu и памяти, оптимизировать код с учетом этой информации. Написание самого быстрого решения не является целью задания.

Для генерации графа вам понадобится graphviz. Для пользователей windows не забудьте добавить его в PATH чтобы была доступна команда dot.

Рекомендую внимательно прочитать доп. материалы на русском - там ещё много примеров оптимизации и объяснений как работать с профайлером. Фактически там есть вся информация для выполнения этого задания.

Есть с десяток мест где можно оптимизировать.

Для выполнения задания необходимо чтобы один из параметров (ns/op, B/op, allocs/op) был быстрее чем в BenchmarkSolution (fast < solution) и ещё один лучше BenchmarkSolution + 20% (fast < solution * 1.2), например (fast allocs/op < 10422*1.2=12506).

По памяти (B/op) и количеству аллокаций (allocs/op) можно ориентироваться ровно на результаты BenchmarkSolution ниже, по времени (ns/op) - нет, зависит от системы.

Для этого задания увеличено количество проверок с 3-х до 5 за 8 часов.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

Результат в fast.go в функцию FastSearch
(изначально там то же самое что в SlowSearch).

Пример результатов с которыми будет сравниваться:

```
$ go test -bench . -benchmem

goos: windows

goarch: amd64

BenchmarkSlow-8 10 142703250 ns/op 336887900 B/op 284175 allocs/op

BenchmarkSolution-8 500 2782432 ns/op 559910 B/op 10422 allocs/op

PASS

ok coursera/hw 3.897s
```

Запуск:

go test -v - чтобы проверить что ничего не сломалось

go test -bench . -benchmem - для просмотра производительности

Советы:

- Смотрите где мы аллоцируем память
- Смотрите где мы накапливаем весь результат, хотя нам все значения одновременно не нужны
- Смотрите где происходят преобразования типов, которые можно избежать
- Смотрите не только на графе, но и в pprof в текстовом виде (list FastSearch) - там прямо по исходнику можно увидеть где что
- Задание предполагает использование easyjson. На сервере эта библиотека есть, подключать можно. Но сгенерированный через easyjson код вам надо поместить в файл с вашей функцией
- Можно сделать без easyjson

Примечание:

- easyjson основан на рефлекции и не может работать с пакетом main. Для генерации кода вам необходимо вынести вашу структуру в отдельный пакет, сгенерить там код, потом забрать его в main