Министерство образования и науки РФ

Курганский Государственный Университет

Кафедра БИАС

Отчет

по лабораторной работе №3

по теме: **«Разработка сервера высоконагруженных приложения на Go»**

Выполнил: студент группы Т-309.13

Бутенко А.Ю.

Проверил: Подкорытов Д.А.

## Курган 2016

# **Цель работы:** разработка ПО способного принимать близкое к миллиону количество TCP запросов в секунду.

**Средства:**

1. ПК
2. Язык программирования Go

**Постановка задачи:**

Исходный TCP сервер должен быть в состоянии принять и обработать 1M qps. Для создания такого рода сервера придется воспользоваться встроенными в UNIX ядро системными вызовами.

Используя go­-роутины в качестве очереди, для обработки запросов, получается обработать около 1M qps. При увеличении лимита socket-файлов – OSX получает сигнал ядра “Kernel overflow” и перезагружается.

Используя расширенное количество socket-файлов и функции ядра readv/writev удалось получить результат, близкий к 2M TCP qps.

В приведенном примере используется STL для Go по работе с net. Именно в ней используются обычные системные вызовы для работы с сетью. И именно она позволяет получить результат близкий к 1M.



1. **package** main
2. **import** (
3. "bufio"
4. "log"
5. "net"
6. "sync"
7. "time"
8. )
9. **var** wg sync.WaitGroup
10. **var** err error
11. **var** cnt int
13. func main() {
14. l, \_ := net.Listen("tcp", ":1234")
15. log.Println(l.Addr().String())
16. wg.Add(1)
17. **go** func() {
18. **defer** wg.Done()
19. **for** \_ = **range** time.Tick(time.Second) {
20. log.Println(cnt)
21. cnt = 0
22. }
23. }()
24. **for** {
25. conn, \_ := l.Accept()
26. **go** func() {
27. scanner := bufio.NewScanner(conn)
28. **for** {
29. cnt++
30. **if** ok := scanner.Scan(); !ok {
31. **break**
32. }
33. }
35. conn.Close()
36. }()
37. }
38. }

**Вывод:**

Заданное приложение написано. Задача решена.