



Как писать FastAGI на perl, спиться и перейти на GO



Промокодики



Zadarma

AsterZadarma21



ASTERCONF'2021

Рейтинг современных ЯП



Mar 2021	Mar 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	15.33%	-1.00%
2	1	▼	Java	10.45%	-7.33%
3	3		Python	10.31%	+0.20%
4	4		C++	6.52%	-0.27%
5	5		C#	4.97%	-0.35%
6	6		Visual Basic	4.85%	-0.40%
7	7		JavaScript	2.11%	+0.06%
8	8		PHP	2.07%	+0.05%
9	12	▲	Assembly language	1.97%	+0.72%
10	9	▼	SQL	1.87%	+0.03%
11	10	▼	<u>Go</u>	1.31%	+0.03%
12	18	▲	Classic Visual Basic	1.26%	+0.49%
13	11	▼	R	1.25%	-0.01%
14	20	▲	Delphi/Object Pascal	1.20%	+0.48%
15	36	▲	Groovy	1.19%	+0.94%
16	14	▼	Ruby	1.18%	+0.13%
17	17		<u>Perl</u>	1.15%	+0.24%
18	15	▼	MATLAB	1.04%	+0.05%
19	13	▼	Swift	0.95%	-0.28%
20	19	▼	Objective-C	0.91%	+0.17%



Стек Perl технологий для работы с FastAGI



- FastAGI server
- Net::Server
- fork/prefork

Почему Go



- Набирает популярность, легко начать изучать
- Компилируемый, быстрее интерпретируемых
- Статическая типизация
- Есть сборщик мусора
- Умеет многопоточность из коробки
- C - family
- Удобный и минималистичный синтаксис

Почему Go



- Goroutines
 - легковесные threads
 - дешево в разработке
 - goroutines часть OS threads
- Channels
 - общение между goroutines
 - синхронизация состояний и т.д.



Простейший пример на Go



```
test.go x
1 package main
2
3 import "fmt"
4
5 func main() {
6     var msg = "Hello Asterconf"
7
8     for i := 0; i <= 10; i++ {
9         fmt.Println(msg, 2021+i)
10    }
11 }
```

инструменты GO



- go
 - test
 - fmt
 - build
 - install
- Отладка
 - gdb
- менеджер пакетов

```
package math
```

```
import "testing"
```

```
func TestAverage(t *testing.T) {  
    var v float64  
    v = Average([]float64{1,2})  
    if v != 1.5 {  
        t.Error("Expected 1.5, got ", v)  
    }  
}
```

```
go test
```

```
$ go test  
PASS  
ok      golang-book/chapter11/math  0.032s
```

AGI пакет



- поддержка AGI и FastAGI
- простой знакомый интерфейс
- лицензия BSD
- инсталляция
 - `go get github.com/zaf/agi`

AGI пакет



- AGI переменные
 - Env map[string] string
- Структура ответов
 - цифровой ответ
 - Res int
 - дополнительные данные
 - Dat string

AGI пакет



- Примеры команд и методов
 - `func (a *Session) Answer() (Reply, error)`
 - `func (a *Session) Hangup(channel ... string) (Reply, error)`
 - `func (a *Session) SayDigits(digit int, escape string) (Reply, error)`

Использование



- Создание новой сессии
 - `myAGI := agi.New()`
- инициализация чтения AGI окружения
 - `myAGI.Init(nil)`
- выполнение команд
 - `myAGI.StreamFile("hello-world", "#")`

Пример AGI

```
package main

import (
    "log"
    "github.com/zaf/agi"
)

func main() {
    // Create a new AGI session and Parse the AGI environment.
    myAgi := agi.New()
    err := myAgi.Init(nil)
    if err != nil {
        log.Fatalf("Error Parsing AGI environment: %v\n", err)
    }
    // Print a message on the asterisk console using Verbose.
    _, err := myAgi.Verbose("Hello World")
    if err != nil {
        log.Fatalf("AGI reply error: %v\n", err)
    }
}
```

Пример FastAGI



```
func main() {
    // Create a tcp listener on port 4573 and start a new goroutine for each connection.
    ln, err := net.Listen("tcp", ":4573")
    if err != nil {
        log.Fatal(err)
    }
    defer ln.Close()
    for {
        conn, err := ln.Accept()
        if err != nil {
            log.Println(err)
            continue
        }
        go connHandle(conn)
    }
}
```



Пример FastAGI

```
func connHandle(c net.Conn) {
    defer c.Close()
    myAgi := agi.New()
    // Create I/O buffers
    rw := bufio.NewReadWriter(bufio.NewReader(c), bufio.NewWriter(c))
    err := myAgi.Init(rw)
    if err != nil {
        log.Println(err)
        return
    }
    rep, err := myAgi.StreamFile("hello", "1234567890#*")
    if err != nil {
        log.Println(err)
        return
    }
    // Check AGI command return value
    if rep.Res == -1 {
        log.Println("Failed to playback file")
    }
}
```

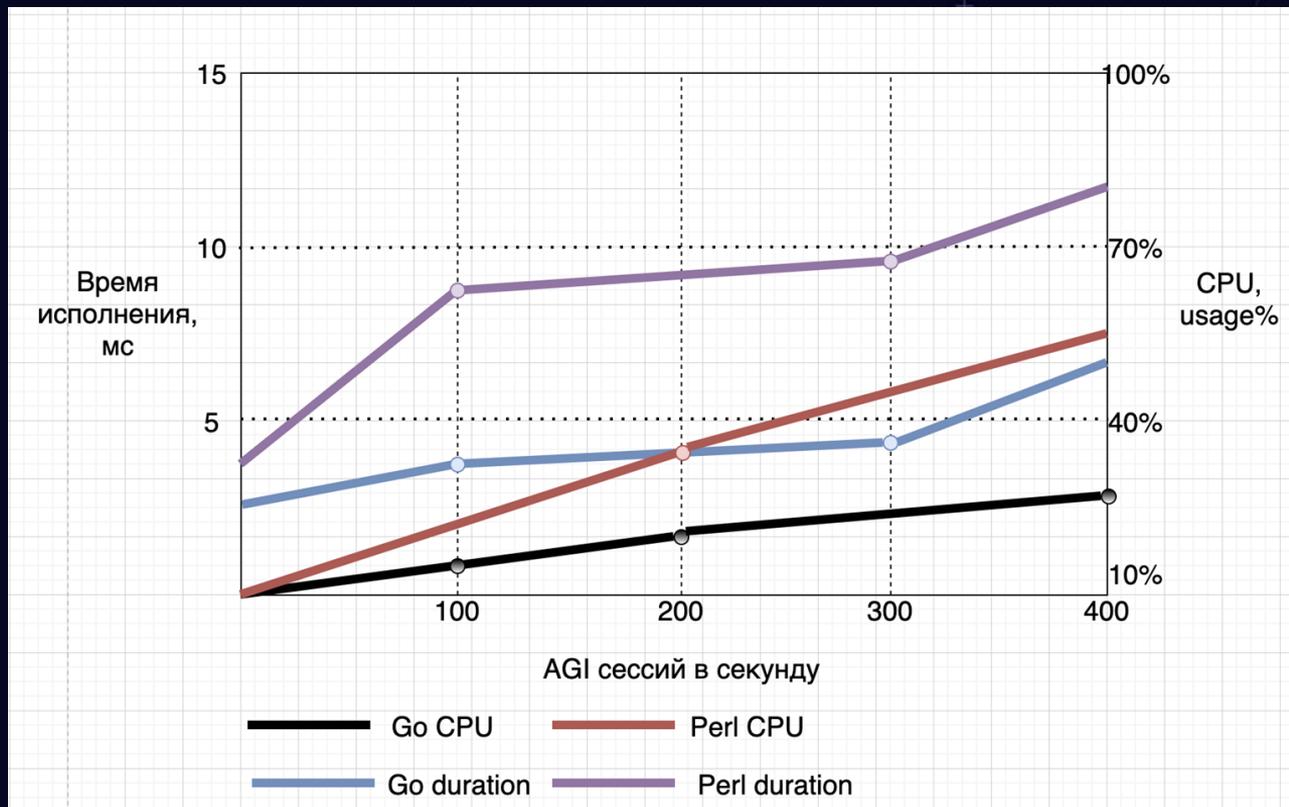
Сравнение производительности Perl и Go



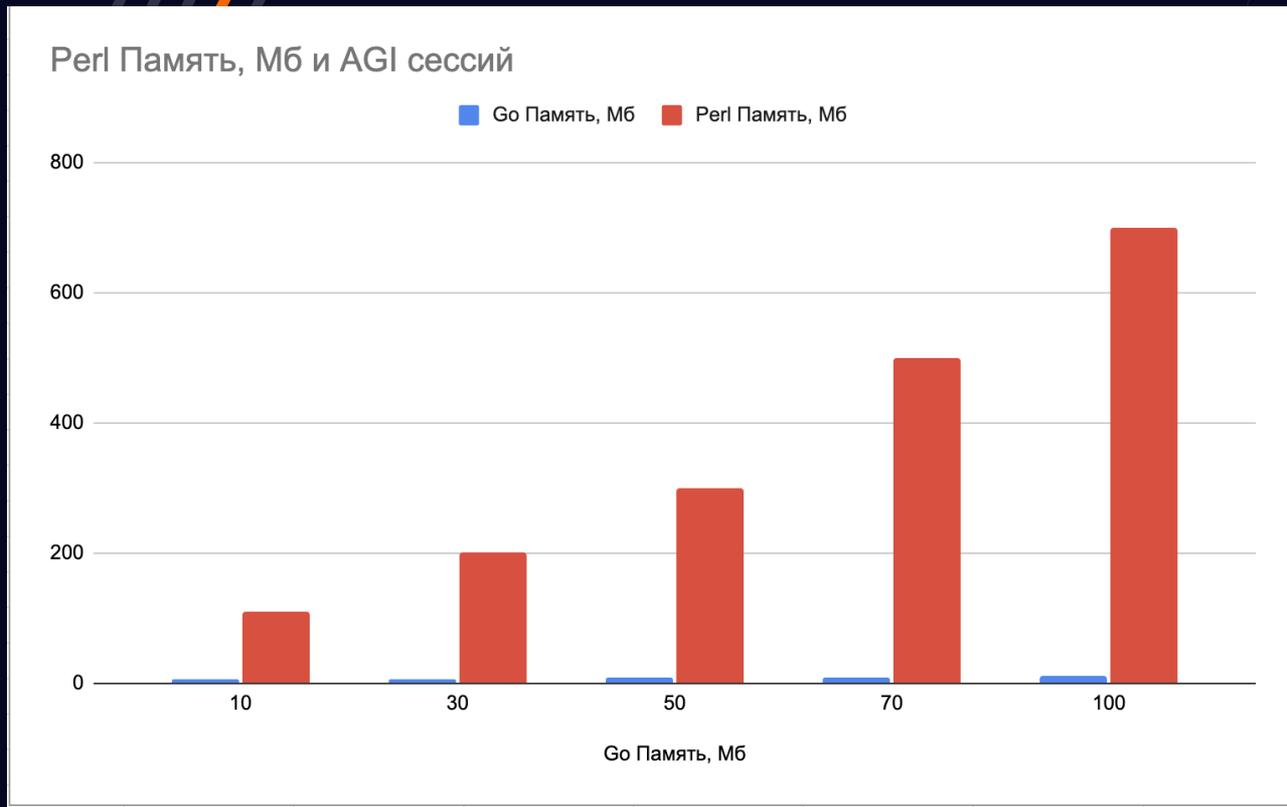
- Perl
 - Asterisk::AGI
 - Net::Server::Prefork
- Go
 - agi пакет
 - net пакет
- генерируем трафик с помощью sipr
 - старт сессии
 - проверка статуса канала
 - answer
 - playback
 - hangup



Сравнение производительности Perl и Go



Сравнение производительности Perl и Go



Спасибо за внимание!

У кого-нибудь есть вопросы?

lagutas@gmail.com
telegram: @lagutas
zadarma.com