Porovnávanie efektivity rôznych jazykov vo výuke programovania

Ján Kelemen

školiteľ: Ing. František Gyárfáš, CSc.

Informácie o mojej bakalárskej práci

- webová aplikácia s rôzne ťažkými úlohami, ktoré sa budú dať programovať v rôznych jazykoch
- používateľské riešenia budú kontrolované pomocou TDD
- podrobné štatistiky o behu programu
- bude možné prehliadať zverejnené riešenia ostatných používateľov a zoradiť ich podľa rýchlosti
- riešenia budú ukladané na serveri v databáze
- používateľ bude môcť vytvoriť konto a pridať svoje vlastné zadanie

Existujúce riešenia

Go

1. Two Sum

Easy | 28253 | 905 | Add to List | Share

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output: [1,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

Constraints:

- `2 <= nums.length <= 10^4`
- `-10^9 <= nums[i] <= 10^9`
- `-10^9 <= target <= 10^9`
- **Only one valid answer exists.**

Follow-up: Can you come up with an algorithm that is less than `O(n^2)` time complexity?

Accepted 5,747,149 | Submissions 11,941,298

```go
func twoSum(nums []int, target int) []int {

}
```

submit a solution

Please insert your source code or choose a file: Choose File No file chosen

```go
package main
import "fmt"

func main(){
    // your code goes here
}
```

Go (go 1.12.1)

Submit!

First language: go Select from solutions: 01.02.22, 10:53:59 click to reset solution Select from test: 01.02.22, 10:53:59 (final) click to reset test

First language:

description

Fizz buzz is a group word game for children to teach them about division. Players take turns to count incrementally, replacing any number divisible by three with the word "fizz", and any number divisible by five with the word "buzz".

solution 1 click to remove

```
package main

import (
	"strconv"
)

func FizzBuzz1_000_000() []string {
	res := make([]string, 0, 1_000_000)
	for i := 1; i <= 1_000_000; i++ {
		if i%3 == 0 && i%5 == 0 {
			res = append(res, "fizzbuzz")
		} else if i%3 == 0 {
			res = append(res, "fizz")
		} else if i%5 == 0 {
			res = append(res, "buzz")
		} else {
			res = append(res, strconv.Itoa(i))
		}
	}
	return res
}
```

test 1 click to remove

```
package main

import "testing"

func TestFizzBuzz1000(t *testing.T) {
	res := FizzBuzz1_000_000()

	tests := []struct {
		name     string
		expecting string
		got      string
	}{
		{name: "on index 14", expecting: "fizzbuzz", got: res[14]},
		{name: "on index 100_000", expecting: "100001", got: res[100_000]},
		{name: "on index 999_999", expecting: "buzz", got: res[999_999]},
	}
	for _, test := range tests {
		t.Run(test.name, func(t *testing.T) {
			if test.expecting != test.got {
				t.Errorf("%s got: %q, expecting: %q", test.name, test.got, test.ex
			}
		})
	}
}
```

Run 1. language | Run and save solution | Run and save test | Run and save both

test OK

compilation time: 0.39 s

real time: 0.02 s

kernel time: 0 s

user time: 0.02 s

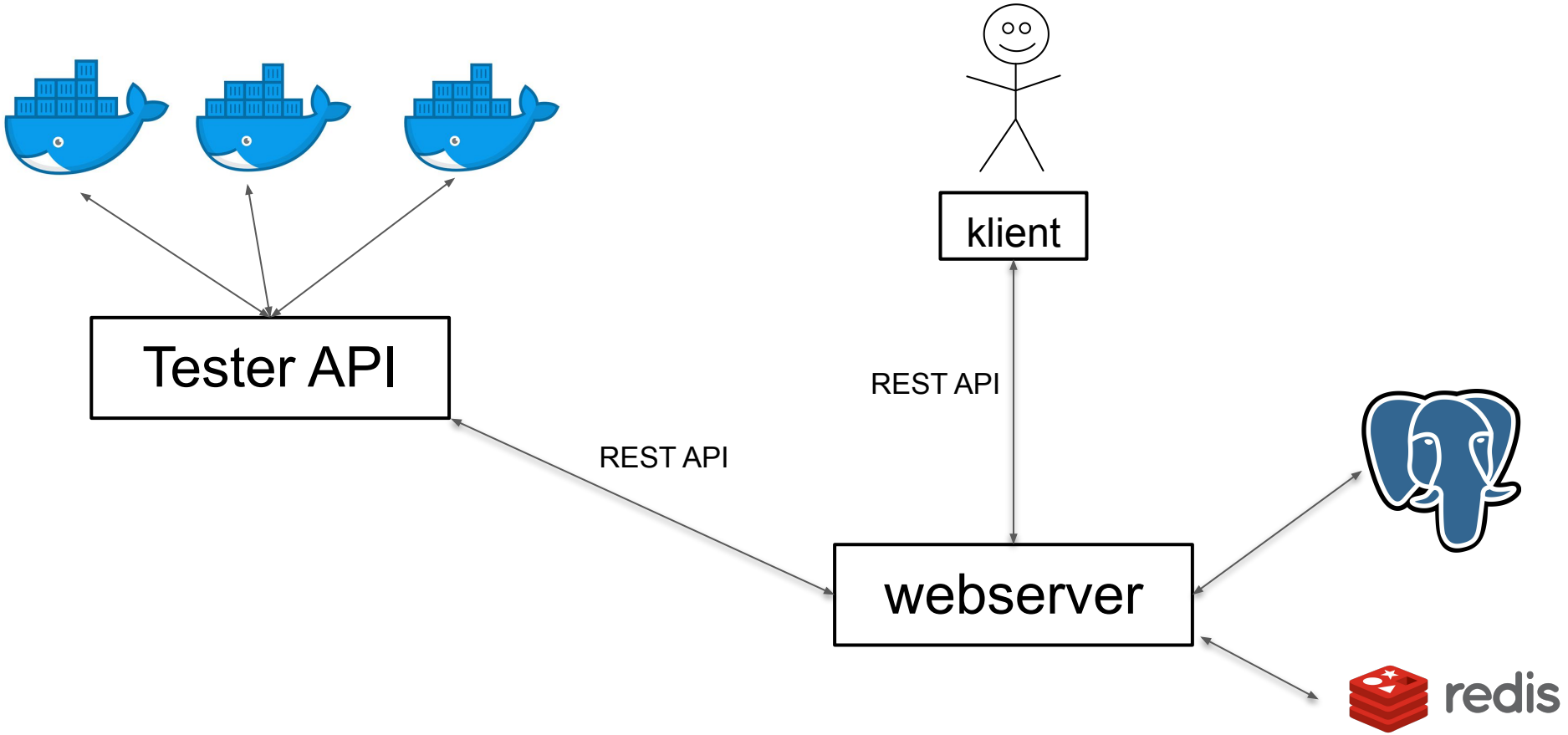
max ram usage: 23.2 mb

binary size: 2.93 mb

test output:

```
=== RUN   TestFizzBuzz1000
=== RUN   TestFizzBuzz1000/on_index_14
=== RUN   TestFizzBuzz1000/on_index_100_000
=== RUN   TestFizzBuzz1000/on_index_999_999
--- PASS: TestFizzBuzz1000 (0.03s)
    --- PASS: TestFizzBuzz1000/on_index_14 (0.00s)
    --- PASS: TestFizzBuzz1000/on_index_100_000 (0.00s)
    --- PASS: TestFizzBuzz1000/on_index_999_999 (0.00s)
PASS
```

Architektúra


klient
Tester API
REST API
REST API
webserver
redis

Ďakujem za pozornosť