

# Ускоряем производительность Python-клиента Redis, используя Golang

---

Букаткин Юрий  
@randomazer

Moscow Python Meetup 58



Программный Регион

Ускоряем производительность Python-клиента Redis, используя Golang



Программный Регион

# Цель

- Быстрый Python-клиент для Redis
- Pipeline, Pooling, многопоточность
- Простое использование в Python
- Читаемость кода



# Альтернативы

- Grumpy
- Gopy
- Go extensions for Python



# Реализация

---



Программный Регион

## Ускоряем производительность Python-клиента Redis, используя Golang

```
package main
```

```
// #cgo pkg-config: python3
// #define Py_LIMITED_API
// #include <Python.h>
// int PyArg_ParseTuple_String(PyObject *, char**, char**, char**, char**);
// int PyArg_ParseTuple_Hashmap_Get_String(PyObject *, char**, char **);
// int PyArg_ParseTuple_Hashmap_Set_String(PyObject *, char**, char **, char **);
// int PyArg_ParseTuple_Connection(PyObject *, char**, long long *);
// int PyArg_ParseTuple_LL(PyObject *, long long *);
// PyObject* Py_String(char *pystring);
import (
    "C"
)
import (
    "bytes"
    "fmt"
    "io"
    "log"
    "net"
    "strconv"
)
```



Ускоряем производительность Python-клиента Redis, используя Golang

```
//export hget
func hget(self *C.PyObject, args *C.PyObject) *C.PyObject {

    var hashmap, key *C.char
    if C.PyArg_ParseTuple_Hashmap_Get_String(args, &hashmap, &key) == 0 {
        return C.PyLong_FromLong(0)
    }

    hashmapStr := C.GoString(hashmap)
    keyStr := C.GoString(key)
    buf := getRawCommand([]string{"hget", hashmapStr, keyStr})
    _, err := cli.sock[0].Write(buf.Bytes())
    if err != nil {
        log.Println(err)
    }
    _ = readBuffer(cli.sock[0])
    buf.Reset()
    return C.PyLong_FromLong(0)
}
```



Ускоряем производительность Python-клиента Redis, используя Golang

```
#define Py_LIMITED_API
#include <Python.h>

PyObject * Connect(PyObject *, PyObject *);
PyObject * add_command(PyObject *, PyObject*);
PyObject * execute(PyObject *, PyObject *);
PyObject * hget(PyObject *, PyObject *);
PyObject * hset(PyObject *, PyObject *);
PyObject * phget(PyObject *, PyObject *);
PyObject * phset(PyObject *, PyObject *);

int PyArg_ParseTuple_Connection(PyObject * args, char**host, long long *port){
    return PyArg_ParseTuple(args, "sL", host, port);
}
```





# Build project

```
.PHONY: build  
build:  
    cd cpipe && go build -buildmode=c-shared -o cpipe.so
```



# Single commands

---



Программный Регион

# Реализация Python

```
import pipe
```

```
cpipe.Connect("localhost", 6379)  
result = cpipe.hset("hashmap", "key", "value")  
result = cpipe.hget("hashmap", "key")
```

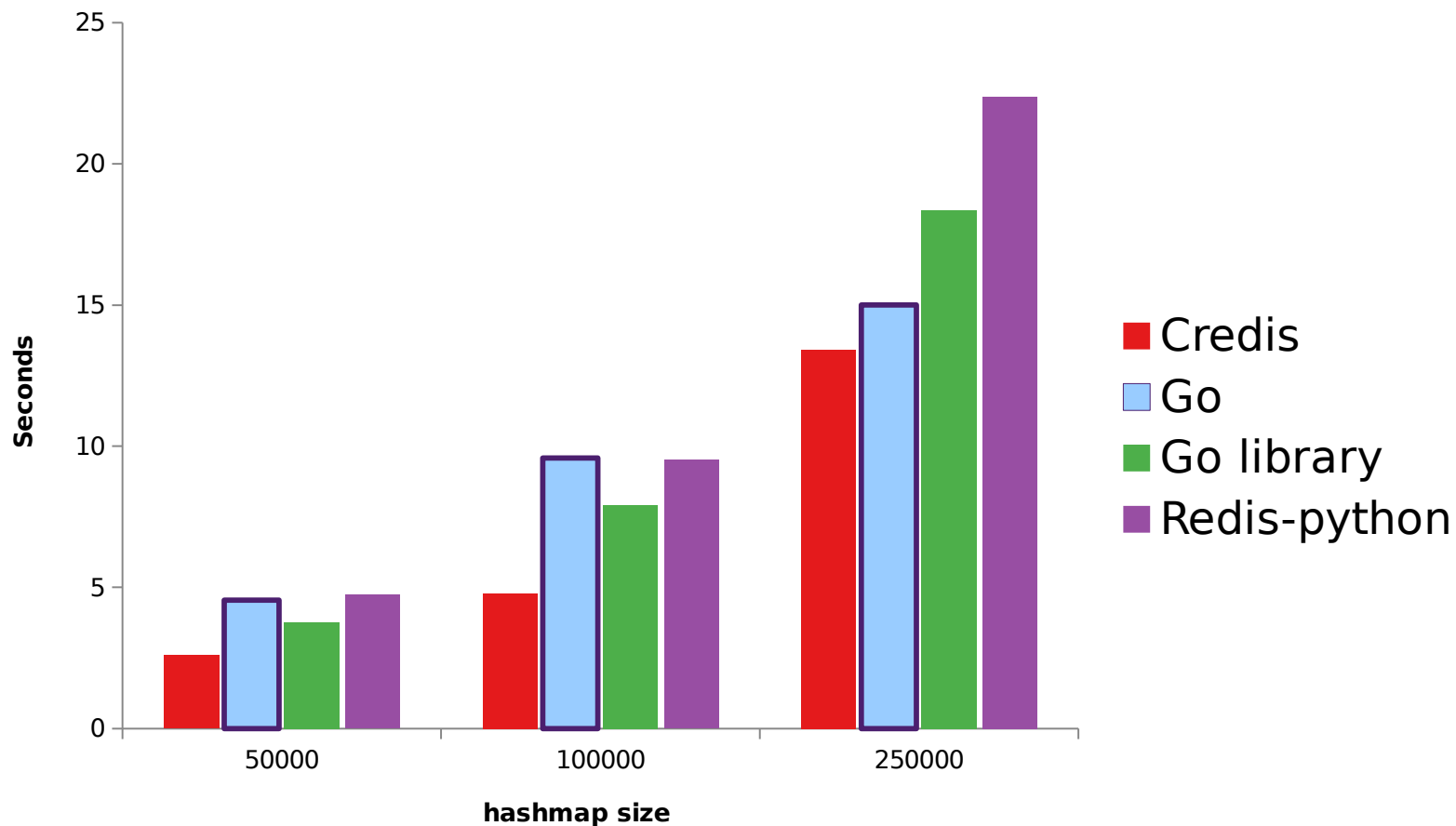


# Тестирование

- Redis-py <https://github.com/andymccurdy/redis-py>
- Credis <https://github.com/yihuang/credis>
- Python + Go (raw realization)
- Python + Go (<https://github.com/go-redis/redis>)



# Benchmarks



# Pipelining

---



Программный Регион

# Структура pipeline

**hset    hashmap    key    value**

**\*4\r\n\$4\r\nhset\$7\r\nhashmap\$3\r\nkey\$5\r\nvalue\r\n**



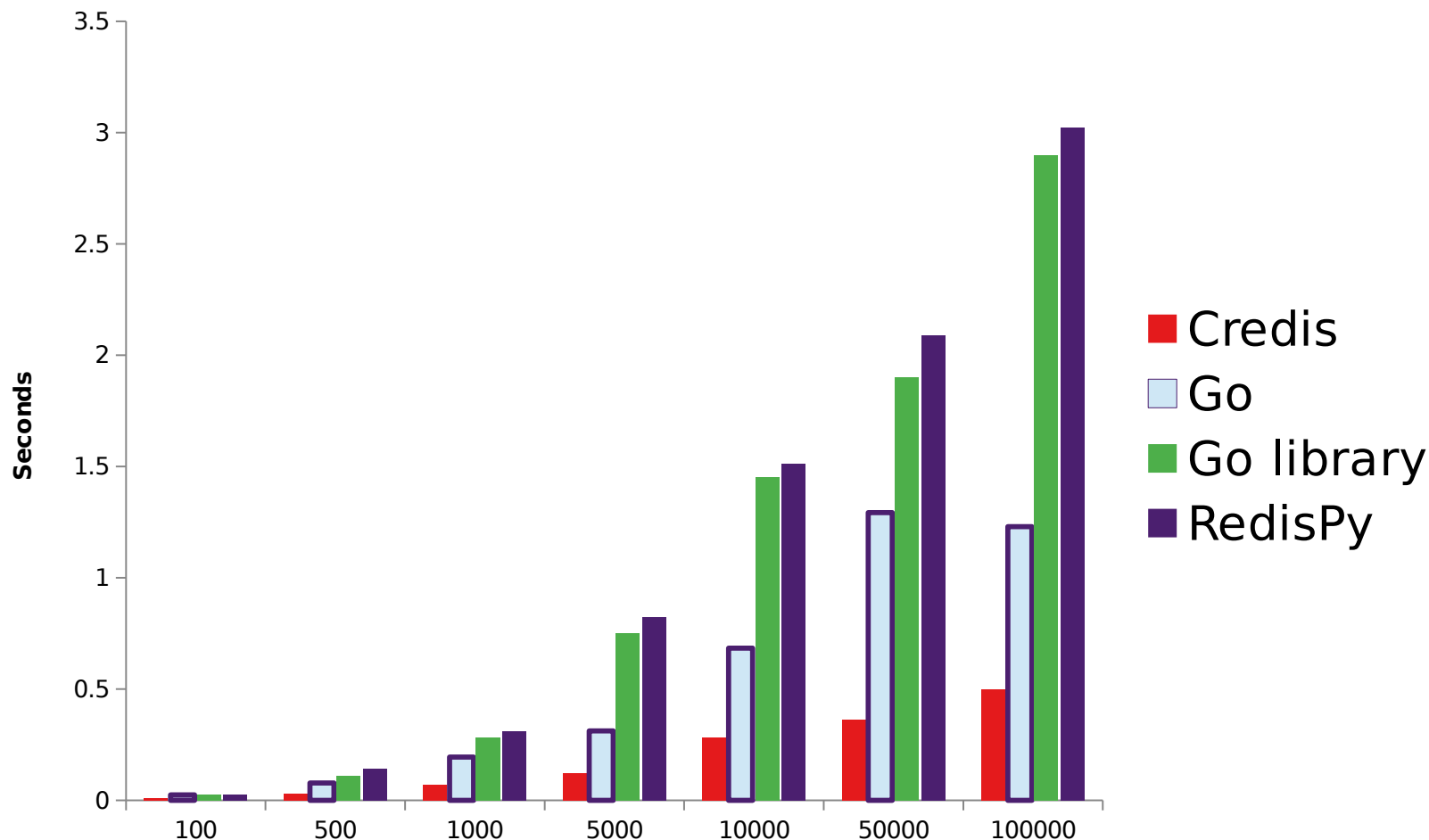
# Реализация Python

```
for key in range(10000):  
    cpipe.add_command("hset", "words", "word|{}".format(key), "1")  
cpipe.execute()
```

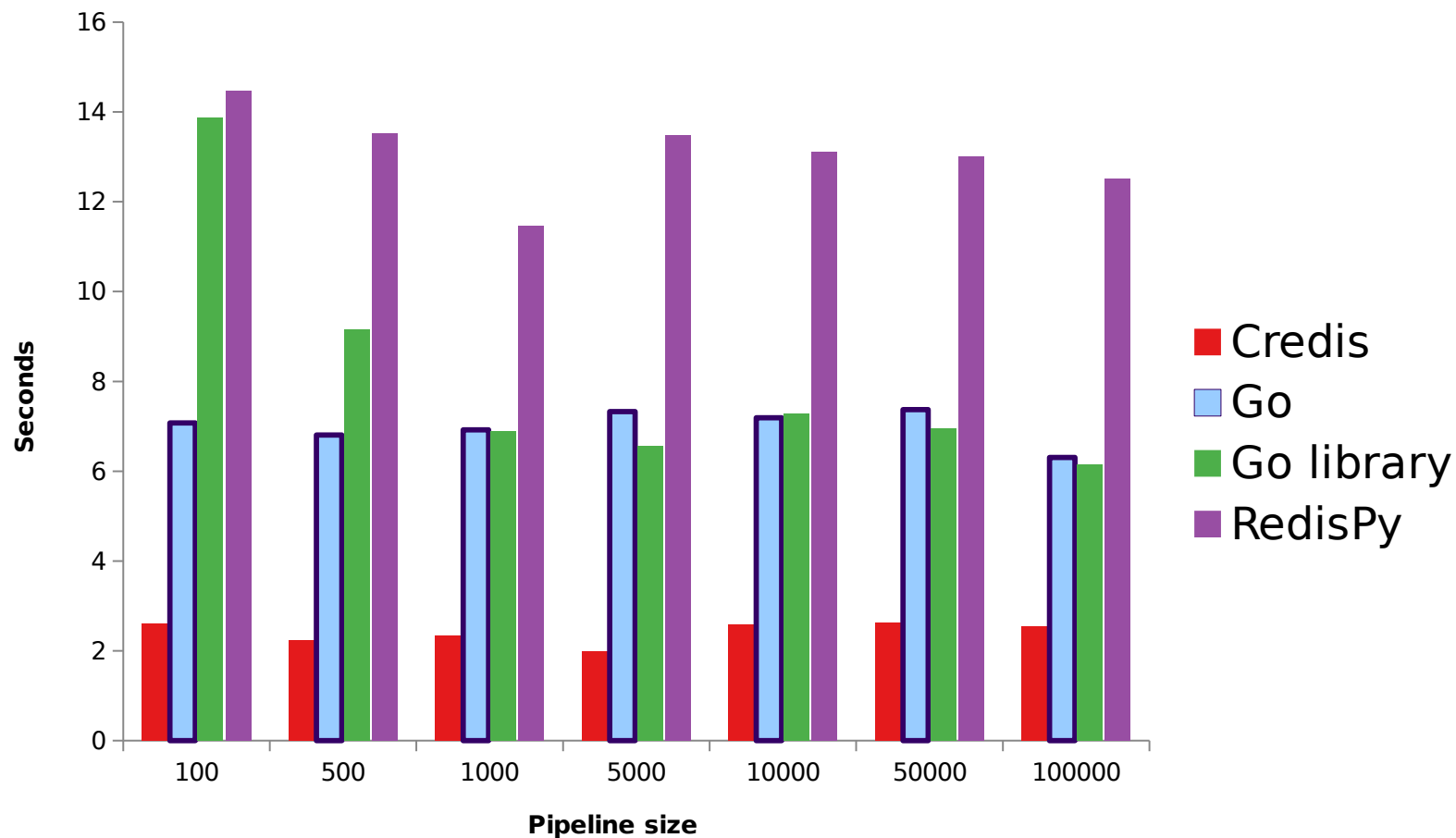




# Benchmarks hashmap size



# Benchmarks pipeline size



# Pooling & Async

---



Программный Регион

# Реализация Python

```
import cpipe

cpipe.ConnectPool(host = "localhost", port = 6379, pool = 10)

for key in range(10000):
    cpipe.phset("words", "word|{}".format(key), "1")
```

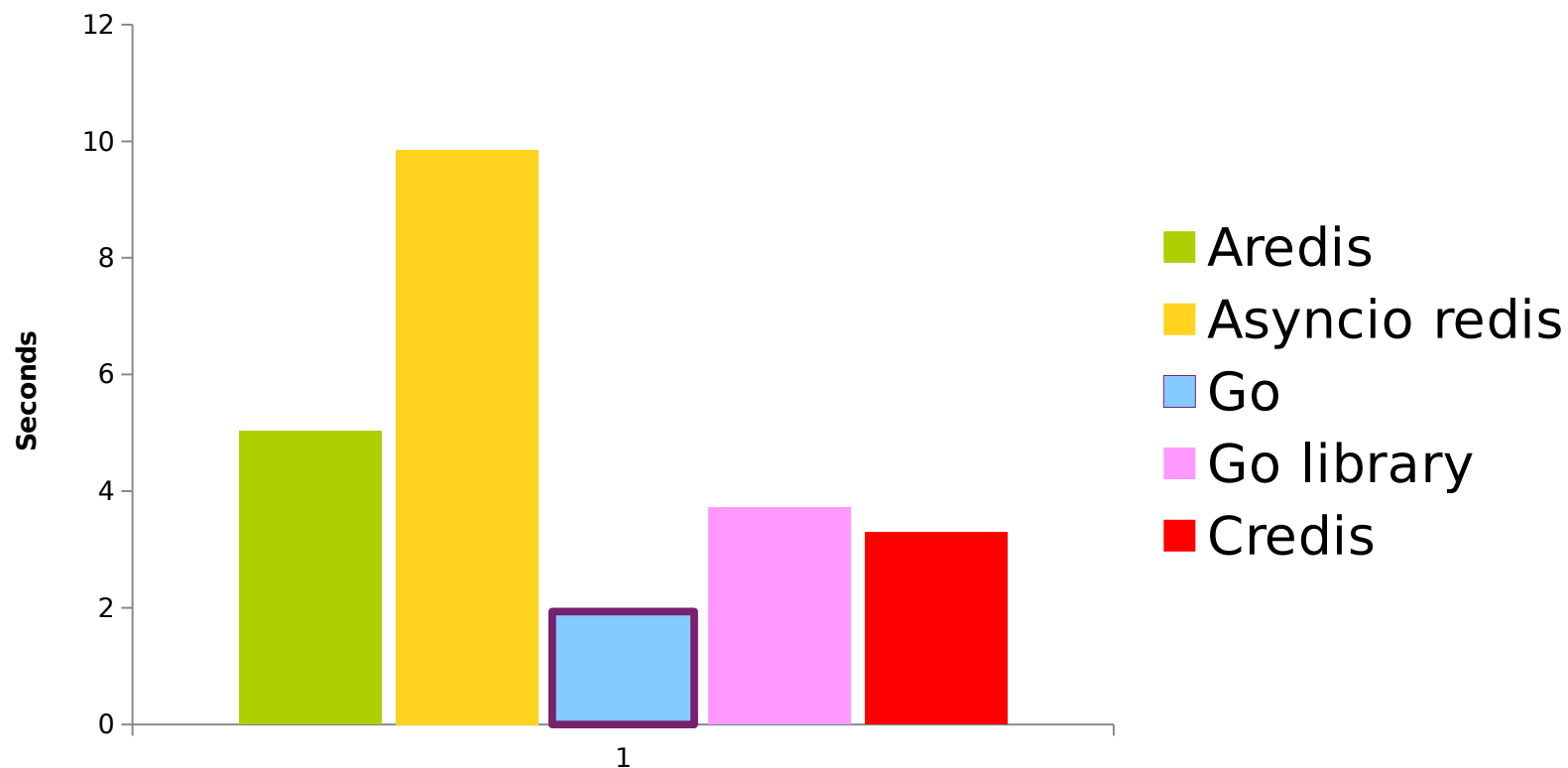


# Тестирование

- Credis <https://github.com/yihuang/credis>
- Asyncio redis <https://github.com/jonathanslenders/asyncio-redis>
- Aredis <https://github.com/NoneGG/aredis>
- Python + Go (raw realization)
- Python + Go-redis library <https://github.com/go-redis/redis>



# Benchmark



# Выводы

---



Программный Регион

# Достоинства

- Ощутимый прирост скорости pipeline и async
- Читаемый код на Go
- Максимально простое использование в Python
- Инкапсуляция всей логики работы библиотеки в Golang





# Недостатки

- Мало информации
- Сложность реализации по сравнению с другими методами
- Вся ответственность за результат лежит на разработчике



# Полезные ссылки

- <https://github.com/deslum/gopipe>
- <http://nigerlittlepoole.com/post/153224915028/using-go-to-boost-python-performance>
- <https://hackernoon.com/extending-python-3-in-go-78f3a69552ac>
- <https://www.datadoghq.com/blog/engineering/cgo-and-python/>
- <https://habr.com/company/mailru/blog/324250/>



# Контакты

**Email:** [Randomazer@gmail.com](mailto:Randomazer@gmail.com)

**Github:** <https://github.com/deslum>

**Twitter:** <https://twitter.com/randomazer>

