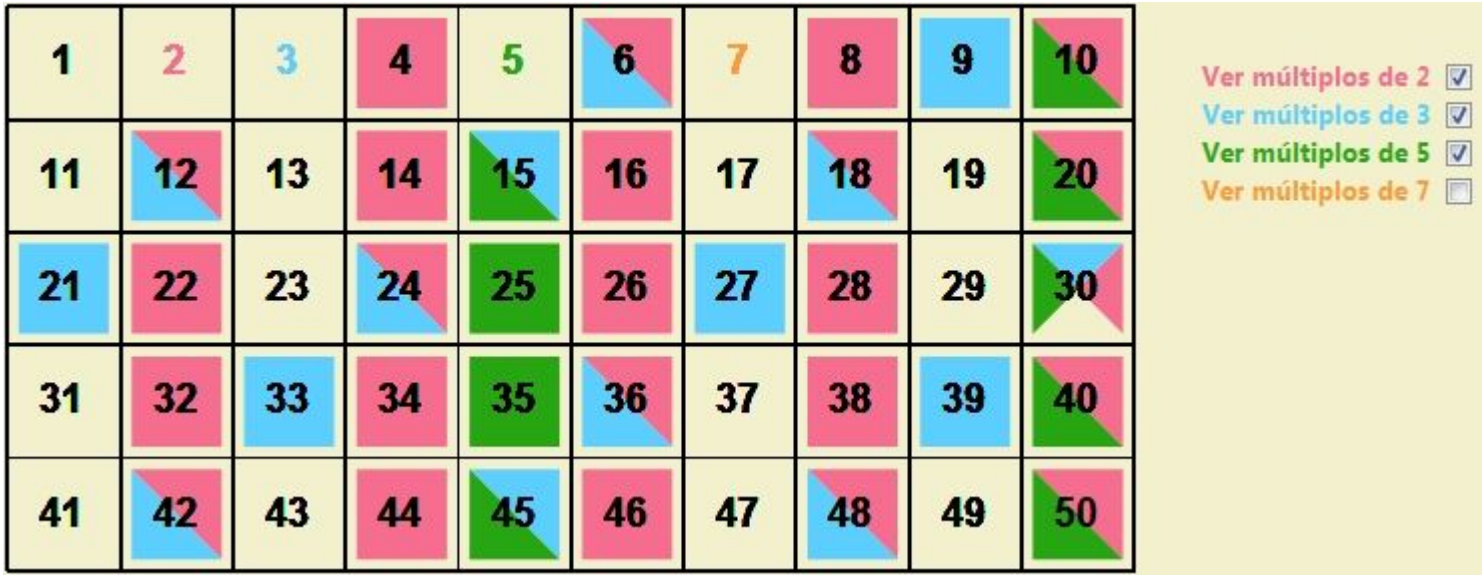
Exercício 05

Equipe 07

Centro de Informática UFPE

Universidade Federal de Pernambuco

Crivo de Eratóstenes



Definir o Objetivo da Avaliação

Comparar o desempenho (RTT) de uma aplicação cliente/servidor usando dois mecanismos de serialização diferentes,Go RPC (HTTP) e gRPC.

Listar os serviços do sistema

Servidor que recebe uma mensagem e retorna-a para o cliente.

Métricas de Desempenho

- Tempo total de execução do Cliente.

- Cliente executa N invocações ao servidor, onde o valor de N variou entre 100, 1000 e 10000.

Listar parâmetros

Parâmetro do Sistema	**Valor**
Hardware	Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz 2.71 GHz, 32,0 GB RAM, x64
Sistema operacional	Microsoft Windows 11
Linguagem de programação	Go
Interfaces de rede	Ligadas
Fonte de alimentação	Rede elétrica
Processos em execução	Apenas os estritamente necessários à realização do experimento

Escolher Fatores

Fator	Nível
Implementação com	goRPC (HTTP), gRPC
Número de Invocações ao servidor	100, 1000, 10000

Técnica de Avaliação

Medição

Projetar o experimento

Os experimentos serão realizados variando o modo de implementação, que irão variar entre Go RPC (HTTP) e gRPC, e variando o número de invocações realizadas ao servidor, que irão variar entre 100, 1000 e 10000 requisições.

Implementação

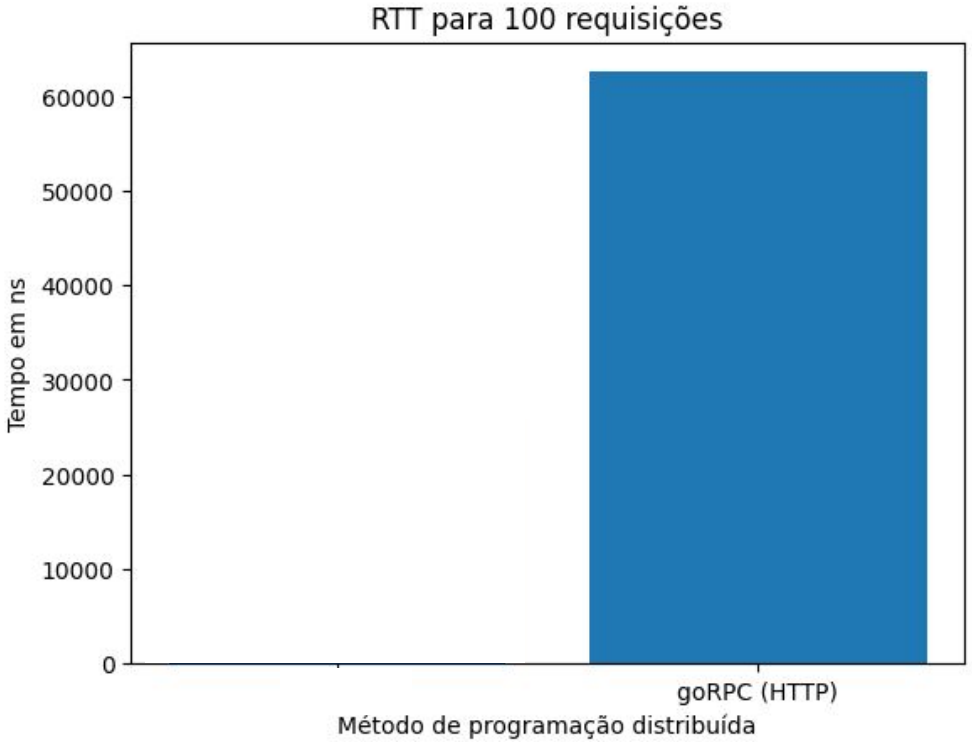
```go
req := shared.Request{Number: 10}
rep := shared.Reply{}
var durations []int64

for i := 0; i < 100; i++ {
    t1 := time.Now()

    err = client.Call("CrivoDeEratostenes.InvocaCrivoDeEratostenes", req, &rep)

    t2 := time.Now().Sub(t1).Nanoseconds()
    durations = append(durations, t2)

    fmt.Printf("RTT: %v: %v\n", t2, rep)
}

mean := calculateMean(durations)
fmt.Printf("Mean duration: %v ns\n", mean)
```

goRPC (HTTP)

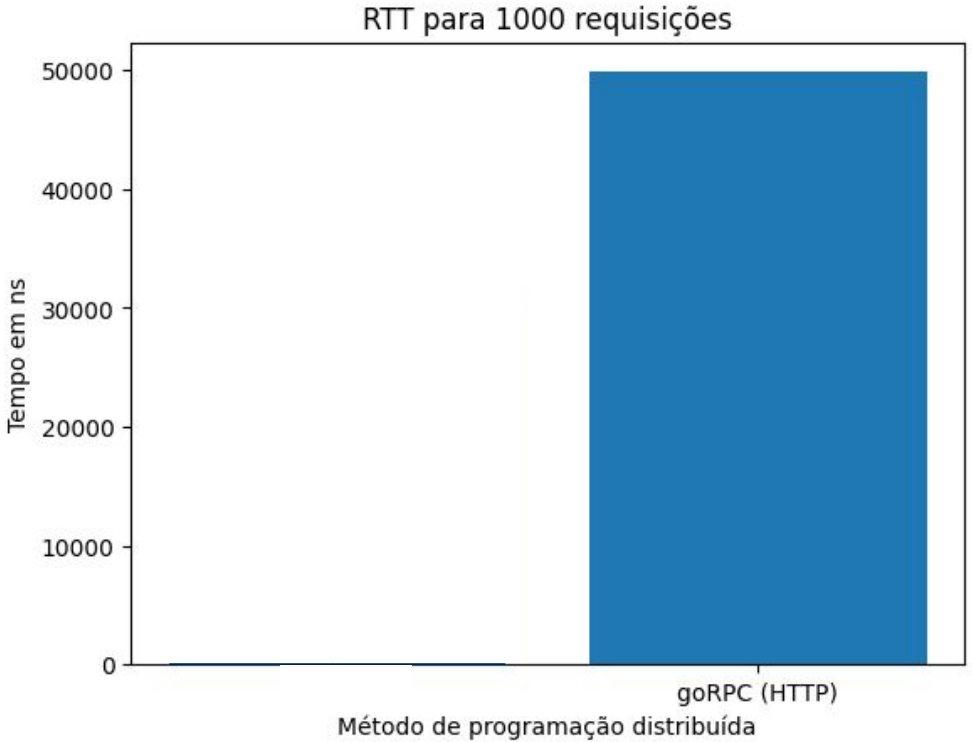
```go
func main() {

    // Estabelece conexão com o servidor
    opt := grpc.WithTransportCredentials(insecure.NewCredentials())
    endPoint := "localhost" + ":" + strconv.Itoa(shared.GrpcPort)
    conn, err := grpc.Dial(endPoint, opt)
    shared.ChecaErro(err, "Não foi possível se conectar ao servidor em"+endPoint)

    // fecha conexões
    defer conn.Close()

    // cria um proxy
    crivo := gen1.NewCrivoClient(conn)

    // cria um contexto para execução remota
    ctx, cancel := context.WithTimeout(context.Background(), time.Second)
    defer cancel()

    var i int32
    for i = 0; i < shared.SampleSize; i++ {
        // invoca operação remota
        reqCrivo := gen1.Request{P1: i}

        repCrivo, err := crivo.Crivo(ctx, &reqCrivo)
        shared.ChecaErro(err, "Erro ao invocar a operação remota.")

        fmt.Printf("Add(%v,%v)=%v", reqCrivo.P1, repCrivo.Primes)
    }
}
```

gRPC

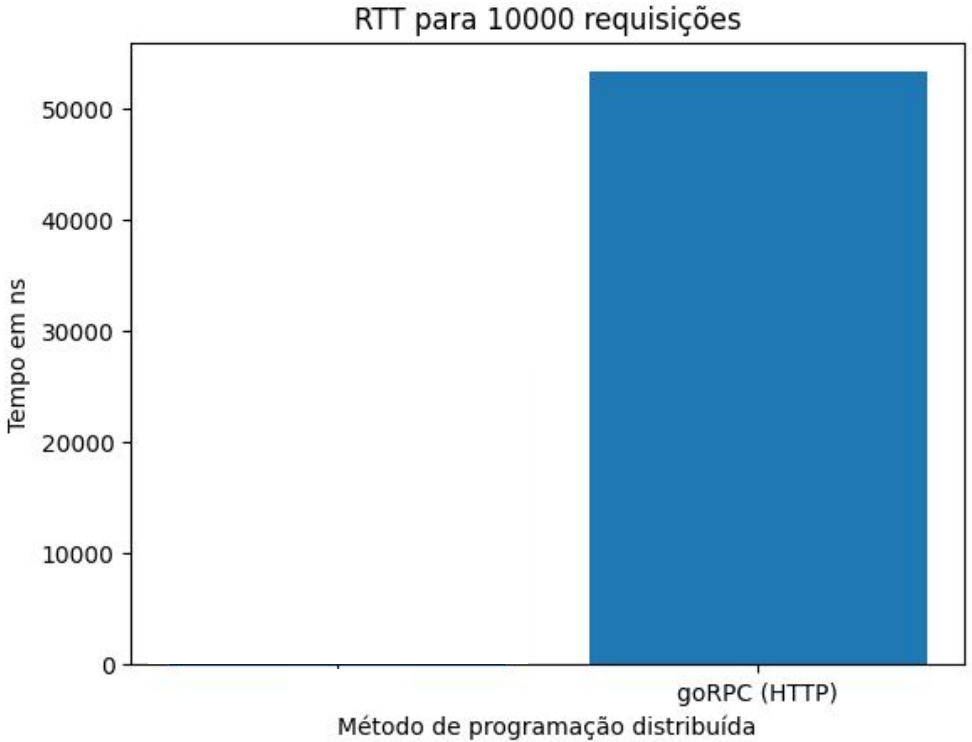
Análise de desempenho



Análise de desempenho



Análise de desempenho



Análise de desempenho

Tivemos problemas com a implementação do gRPC, onde estávamos recebendo alguns erros relacionados a criação dos arquivos *.pb.go.