

tainah.silva@ccc.ufcg.edu.br	<a href="https://github.com/tainahemmanuele/programacao_concorrencia">https://github.com/tainahemmanuele/programacao_concorrencia</a>	JOAO MARCOS ARISTIDES ALMEIDA	SAULO SAMUEL FERREIRA DO NASCIMENTO	TAINAH EMMANUELE SILVA
------------------------------	---	-------------------------------	-------------------------------------	------------------------

## NOTA 6.75

### Q1 - 2.5/3.0

- a (-0,25)
  - funciona
  - complicação desnecessária: um canal (start) que faz todas as goroutines esperarem a criação das demais que sejam iniciadas. Isso é feito num loop com select que sempre executará o caso default; portanto, desnecessário. Canal done desnecessário
- b (-0,25)
  - funciona
  - mesmos problemas da alternativa anterior

### Q2 - 2.25/3.0

- a (-0,25)
  - funciona.
  - mesmos problemas dos programas da questão anterior.
  - poderia ter sido mais simples caso tivesse usado a função `time.Tick(time.Duration(tempo))` discuta em sala
- b (-0,5)
  - funciona
  - mesmos problemas da q1 e da alternativa anterior
  - ruim, usou valores default para controlar a comunicação das goroutines, ao invés das primitivas do canal (`num > 0`)

### Q3 - 2.0/4.0

- experimento ruim. qual a razão de separar em múltiplos plots quando se quer comparar o resultado da variação de um único fator (quantidade de threads)?
- resultados estranhos. p.ex consumo de memória caiu quando compararmos os resultados com 32 e 64 threads.
- não plotou variabilidade dos resultados

marcus.tenorio@ccc.ufcg.edu.br	<a href="https://github.com/dialmformart/concprog">https://github.com/dialmformart/concprog</a>	MARCUS ANTONIO ROCHA TENORIO	KAIO KASSIANO MOURA OLIVEIRA	
--------------------------------	---	------------------------------	------------------------------	--

## NOTA 6.5

### Q1 - 3.0/3.0

- a - ok

- b - ok

#### Q2 - 1.5/3.0

- a - ruim. busy wait?
- b - ruim. busy wait?

#### Q3 - 2.0/4.0

- plots ruins. só consideraram duas threads/goroutines. escala ruim no eixo y. Os resultados são determinísticos? não deveria plotar a variabilidade dos resultados (ou indicar que não são variáveis?) seria importante entender como cresce o consumo conforme se aumenta o número de threads/goroutines

mattheus.rodriques@ccc.ufcg.edu.br	<a href="https://github.com/MattheusB/paradigma-concorrente">https://github.com/MattheusB/paradigma-concorrente</a>	ANARCO QUARESMA ZEFERINO NASCIMENTO	JOSE MANOEL DOS SANTOS FERREIRA	MATTHEUS BRITO RODRIGUES
------------------------------------	---	--	---------------------------------------	--------------------------------

#### NOTA 4.5

#### Q1 - 3.0/3.0

- a - funciona
- b - funciona

#### Q2 - 1.0/3.0

- a - ruim. pq não usar um select para o timeout? pq verificar o tempo limite nas funções request ao invés de uma única vez no gateway
- b - ruim. pq não usar um select para o timeout?

#### Q3 - 0.5/4.0

- ruim. Relatório não reporta experimento direito. Foram rodadas mais de uma execução?

hiago.sousa@ccc.ufcg.edu.br	<a href="https://github.com/hiagofnfs/lista01-progconcorrente">https://github.com/hiagofnfs/lista01-progconcorrente</a>	HIAGO NATAN FERNANDES DE SOUSA	GABRIELA MOTTA OLIVEIRA
-----------------------------	---	--------------------------------------	----------------------------

#### NOTA 5.0

#### Q1 - 1.5/3.0

- a - ok
- b - errado. os requests executam de modo serial

#### Q2 - 1.5/3.0

- a - quase ok. o timeout poderia estar na função gateway
- b - errado. os requests executam de modo serial

#### Q3 - 2.0 /4.0

- relatório ruim. não mostra variabilidade.

- dados estranhos. o consumo de memória em java cai de 10 para 50? além disso, a diferença de consumo entre go e java está grande demais

---

amanda.costa@ccc.ufcg.edu.br	<a href="https://github.com/dalesEwerton/PC-Lista1">https://github.com/dalesEwerton/PC-Lista1</a>	AMANDA VIVIAN ALVES DE LUNA E COSTA	AMINTAS VICTOR RAMOS PEREIRA	DALES EWERTON LOPES FRAGOSO
------------------------------	---	-------------------------------------	------------------------------	-----------------------------

## NOTA 8.5

### Q1 - 3.0/3.0

- a - ok
- b - ok

### Q2 - 1.5/3.0

- a - ruim. pq coordenar goroutines com valores adhoc no canal ao invés de usar um select + timeout
- b - ruim. pq coordenar goroutines com valores adhoc no canal ao invés de usar um select + timeout

### Q3 - 4.0/4.0

- ok
- 

jose.nunes@ccc.ufcg.edu.br	<a href="https://github.com/Benardi/concurrent_prog">https://github.com/Benardi/concurrent_prog</a>	JOSE BENARDI DE SOUZA NUNES	GUSTAVO DINIZ MONTEIRO	RUAN ROBERTO ELOY SILVEIRA
----------------------------	---	-----------------------------	------------------------	----------------------------

## NOTA 8.5

### Q1 - 2.5/3.0

- a - ok +- qual o sentido de ter um select com somente um case?
- b - ok +- qual o sentido de ter um select com somente um case?

### Q2 - 2.0/3.0

- a - deveria ser baseado em CSP não em locks
- b - ok

### Q3 - 4.0/4.0

- ok
- 

filipe.lima@ccc.ufcg.edu.br	<a href="https://github.com/filipegl/lista-1-prog-concorrente">https://github.com/filipegl/lista-1-prog-concorrente</a>	FILIPPE GOMES DE LIMA	JOSE THIAGO DOS SANTOS SILVA	LUCAS BARROS ROCHA
-----------------------------	---	-----------------------	------------------------------	--------------------

## NOTA 8.25

### Q1 - 2.75/3.0

- a - ok
- b - ok +- muito complicado. não precisaria de um waitgroup

### Q2 - 2.5/3.0

- a - ok +- muito complicado poderia ter os temporizadores da biblioteca padrão
- b - ok +- muito complicado poderia ter os temporizadores da biblioteca padrão

### Q3 - 3.0/4.0

- ok. bons experimentos mas me parece que há algo estranho nos dados obtidos, principalmente nos que usam java. Os programas deveriam ser mais simples (deveriam fazer nada, para obtermos o custo somente das goroutines/threads ao invés dos programas)

---

vinicius.jorge.silva@ccc.ufcg.edu.br	<a href="https://github.com/viniciusjps/pc-lista1">https://github.com/viniciusjps/pc-lista1</a>	VINICIUS JORGE PEREIRA DA SILVA	HEMILLAINY SUELLEN SOUSA SANTOS	SAMMARA BESERRA NUNES
--------------------------------------	---	---------------------------------	---------------------------------	-----------------------

## NOTA 5.5 + 0.3

### Q1 - 3.0/3.0

- a - ok
- b - ok

### Q2 - 2.0/3.0

- a - ok
- b - ruim. poderiam ter usado um timeout no select

### Q3 - 0.5/4.0

- relatório muito ruim. pq não plotaram os resultados ao invés de mandar screenshots?

---

felipe.calixto@ccc.ufcg.edu.br	<a href="https://github.com/felipeemerson/prog-conc-lista1">https://github.com/felipeemerson/prog-conc-lista1</a>	FELIPE EMERSON DE OLIVEIRA CALIXTO	WESLEY LUCENA QUEIROZ	
--------------------------------	---	------------------------------------	-----------------------	--

## NOTA 6.25

### Q1 - 1.5/3.0

- a - ok +- pq um select que só tem um case?

- b - muito complicado. pq um select com só um case. pq um goroutine só para saber se o canal já está "cheio"?

**Q2 - 2.75/3.0**

- a - ok
- b - muito complicado. pq um goroutine só para saber se o canal já está "cheio"?

**Q3 - 2.0/4.0**

- +- pq não plotou a variabilidade dos dados
- segundo plot estranho. do que se trata? não foi discutido no relatório

gabriel.almeida.az evedo@ccc.ufcg.edu.br	<a href="https://github.com/cassioegc/lista1-concorrente">https://github.com/cassioegc/lista1-concorrente</a>	GABRIEL ALMEIDA AZEVEDO	CASSIO EDUARDO GABRIEL CORDEIRO	GEOVANE DO NASCIMENTO SILVA
---	---	-------------------------------	--	-----------------------------------

**NOTA 7.5**

**Q1 - 3.0/3.0**

- a - ok
- b - ok

**Q2 - 2.0/3.0**

- a - ok
- b - muito complicado. poderia ter continuado a usar o timeout no select do item anterior

**Q3 - 2.5/4.0**

- plot ruim. escala ruim de ler. poderia ter mostrado variabilidade.
- seria melhor plotar java vs go num mesmo plot. qual a razão na diferença das runtimes
- dados estranho. medições muito altas, tanto para go quanto para java

ronan.souza@ccc.ufcg.edu.br	<a href="https://github.com/yurikelvin/lista_fpc">https://github.com/yurikelvin/lista_fpc</a>	RONAN DE ARAUJO SOUZA	YURI KELVIN MOURA SOUZA E SILVA
-----------------------------	---	--------------------------	------------------------------------

**NOTA 9.5**

**Q1 - 3.0/3.0**

- a - ok
- b - ok

**Q2 - 3.0/3.0**

- a - ok
- b - ok

**Q3 - 3.5/4.0**

- bom. fizeram análises extras que contribuíram para o relatório
- não plotaram a variabilidade

---

david.quaresma@ccc.ufcg.edu.br	<a href="https://github.com/dfquaresma/fpc-lista1">https://github.com/dfquaresma/fpc-lista1</a>	DAVID FERREIRA QUARESMA	RENATO DANTAS HENRIQUES
--------------------------------	---	-------------------------	-------------------------

## NOTA 8.0

### Q1 - 3.0/3.0

- a - ok
- b - ok

### Q2 - 3.0/3.0

- a - ok
- b - ok

### Q3 - 2.0/4.0

- pq só plotaram a mediana? pq sumarizar tanto assim os dados?
  - estranho os resultados do plot para 1 thread em java. dezenas de MB?
- 

lucas.oliveira@ccc.ufcg.edu.br	<a href="https://github.com/LucasFOliveira/pc">https://github.com/LucasFOliveira/pc</a>	LUCAS FERNANDES DE OLIVEIRA	PEDRO HENRIQUE COSTA MAIA	WESLEY SANTOS SILVA
--------------------------------	---	-----------------------------	---------------------------	---------------------

## NOTA 4.0

### Q1 - 2.0/3.0

- a - ok
- b - qual a necessidade das linhas 16-18? Qual a razão do timeout na linha 23? pq um nova goroutine, criada dentro da função request somente para setar um valor?

### Q2 - 1.5/3.0

- a - ok
- b - estranho o sleep da linha 20. seria melhor um select com timeout como foi feito no item anterior. errado, mesmo que o sorteio seja de valores menores que 16, o programa espera 17 segundos

### Q3 - 0.5/4.0

- relatório muito ruim. pq não plotaram os resultados ao invés de mandar screenshots?
-

paulo.felipe.silva@ccc.ufcg.edu.br	<a href="https://github.com/paulofelipefeitosa/concurrent-programming">https://github.com/paulofelipefeitosa/concurrent-programming</a>	FELIPE MOTA DOS SANTOS	PAULO FELIPE FEITOSA DA SILVA
------------------------------------	---	------------------------	-------------------------------

## NOTA 8.0

### Q1 - 3.0/3.0

- a - ok
- b - ok

### Q2 - 1.5/3.0

- a - ok +- esse problema "cheira" select. usar um valor adhoc no canal como flag deveria ser a última opção
- b - a - ok +- esse problema "cheira" select. usar um valor adhoc no canal como flag deveria ser a última opção

### Q3 - 3.5/4.0

- bom relatório, boa análise. por fim, poderia ter plotado resultados menos sumariados
- 

hugo.galvao@ccc.ufcg.edu.br	<a href="https://github.com/KleberCnuto/ListasProConc">https://github.com/KleberCnuto/ListasProConc</a>	HUGO ADDOBBATI GALVAO	KLEBERSON MATHEUS CUNHA SILVA CANUTO	JOSE DE ARIMATEIA MORAIS FILHO	
-----------------------------	---	-----------------------	--------------------------------------	--------------------------------	--

## NOTA 4.5

### Q1 - 2.5/3.0

- a - ok
- b - ok +- linhas 17-19, pq isso?

### Q2 - 1.5/3.0

- a - ruim. pq coordenar goroutines com valores adhoc no canal ao invés de usar um select + timeout
- b - ruim. pq coordenar goroutines com valores adhoc no canal ao invés de usar um select + timeout

### Q3 - 0.5/4.0

- ruim. rodaram somente uma vez os experimentos?
- 

lucas.lima@ccc.ufcg.edu.br	<a href="https://github.com/LucasAurelio/pc-listas">https://github.com/LucasAurelio/pc-listas</a>	LUCAS AURELIO DE LIMA
----------------------------	---	-----------------------

## NOTA 5.5

**Q1 - 3.0/3.0**

- a - ok
- b - ok

**Q2 - 2.5/3.0**

- a - pq nao usar um select (com um temporizador) ao invés de criar uma nova goroutine explícita?
- b - pq nao usar um select (com um temporizador) ao invés de criar uma nova goroutine explícita?

**Q3 - 0.0/4.0**

- não foi entregue

---

thales.souto@ccc.ufcg.edu.br	<a href="https://github.com/AndersonSM/concorrente">https://github.com/AndersonSM/concorrente</a>	THALES HENRIQUE DANTAS SOUTO	ICARO FORTE DANTAS MEDEIROS	ANDERSON SALES DE MENEZES
------------------------------	---	------------------------------------	-----------------------------------	---------------------------------

**NOTA 8.0****Q1 - 2.5/3.0**

- a - select com um case só? qual a lógica disso?
- b - pq vários canais ao invés de um só?

**Q2 - 2.5/3.0**

- a - pq criar uma nova goroutine explícita ao invés de um timer da lib padrão?
- a - pq criar uma nova goroutine explícita ao invés de um timer da lib padrão?

**Q3 - 3.0/4.0**

- seria melhor ter um programa mais simples (talvez um Nop) para não misturar o custo de memória da abstração com o dos programas.
- faltou plotar variabilidade

---

JOSE BRENO DE SOUZA

**NOTA 5.0****Q1 - 2.5/3.0**

- a - ok
- b - select desnecessário

**Q2 - 2.4/3.0**

- a - ok
- b - não para no timeout. depende de uma das goroutines request terminarem

**Q3 - /4.0**