Primeira Lista de Exercíciosde Arquiteturas/PAD

Felipe Menino Carlos

28/04/2020

Objetivo

O principal objetivo desta lista de exercícios é reforçar a fixação dos conceitos vistos em sala de aula sobre pipeline, sobre a lei de Amdahl e sobre o speedup resultante da execução de um programa num sistema paralelo. Para os exercícios nos quais são pedidosgráficos, utilize qualquer programa de plotagem.

Exercícios

Abaixo são apresentados os exercícios propostos na lista, estes foram resolvidos utilizando a linguagem de programação Python junto a biblioteca de funcionalidades para a visualização gráfica plotnine

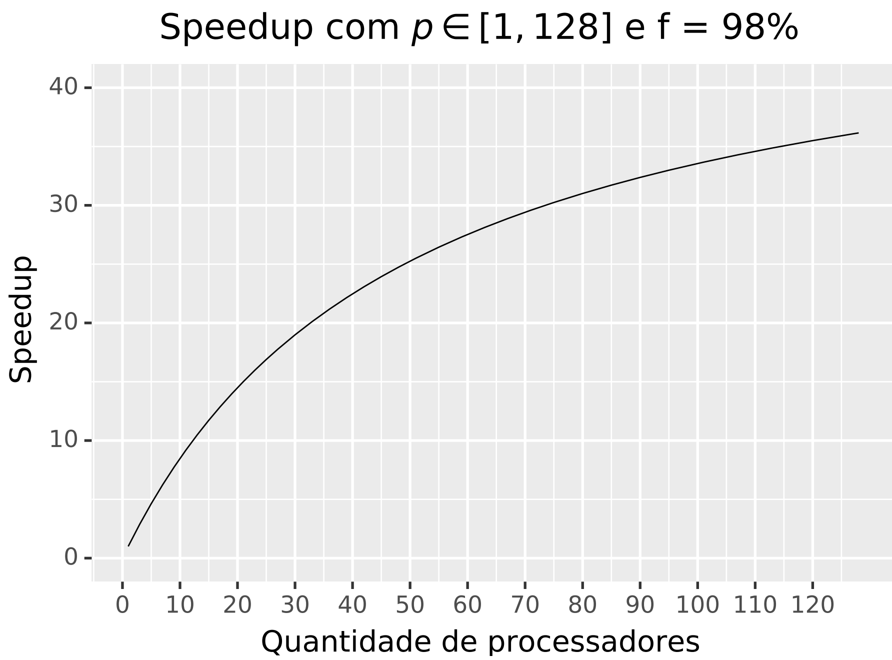
1) Considere uma tarefa que pode ser dividida em sub-tarefas com durações de 15, 25, 30 e 20 segundos, respectivamente. Cada sub-tarefa é executada por um módulo especializado, e a execução é feita em modo pipeline.

- (a) Qual é o tempo de ciclo mínimo para o pipeline ?
- (b) Supondo que existam 100 tarefas a executar, qual o speedup em relação à execução num modo estritamente serial?
- (c) Caso seja possível subdividir umadas sub-tarefas em duas novas sub-tarefas de igual duração, associando um módulo para a execução de cada uma, qual das sub-tarefas deve ser escolhida para divisão?
- (d) Após a divisão proposta no ítem anterior, qual o novo speedup possível em relação à execução estritamente serial das 100 tarefas ?

2) Considere a expressão para o speedup vista em aula, $S_P = \frac{1}{1-f+\frac{f}{P}}$. Plote o speedup como função do número de processadores (P), dentro do intervalo $1 \leq P \leq 128$, supondo que a fração paralelizável (f) de um programa corresponde a:

- (a) 98%

```
## <ggplot: (8754910950069)>
```

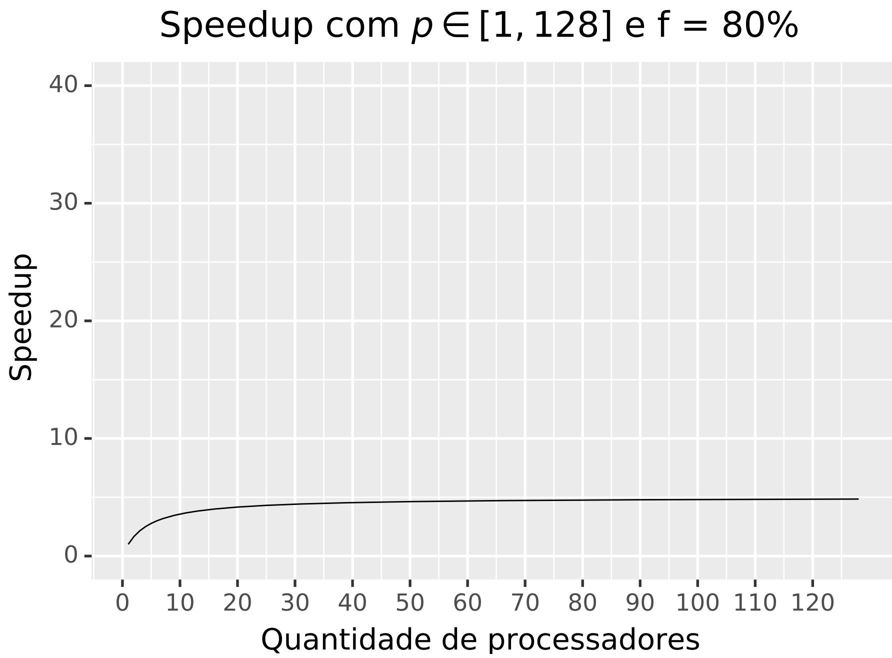


Ao considerar a plotagem realizada, é possível perceber a influência da quantidade de processadores no Speedup quando a fração de paralelização do programa analisado é alto, neste, quanto maior a quantidade de processadores, maior o Speedup.

É importante lembrar que, mesmo com o aumento apresentado ser crescente, há um limite para tal crescimento.

- (b) 80%

```
## <ggplot: (8754885349053)>
```



O resultado desta segunda plotagem ajuda na afirmação feita anteriormente, onde a quantidade de processadores influência positivamente somente quando o programa favorece, ou seja, quando o programa possuí uma boa fração paralelizável.

A função paralelizável pode até ser considerada alta, com 80%, porém, este ainda é um valor baixo para melhor consumir os recursos, o que faz o desempenho não ser muito bom, mesmo com o aumento da quantidade de processadores.

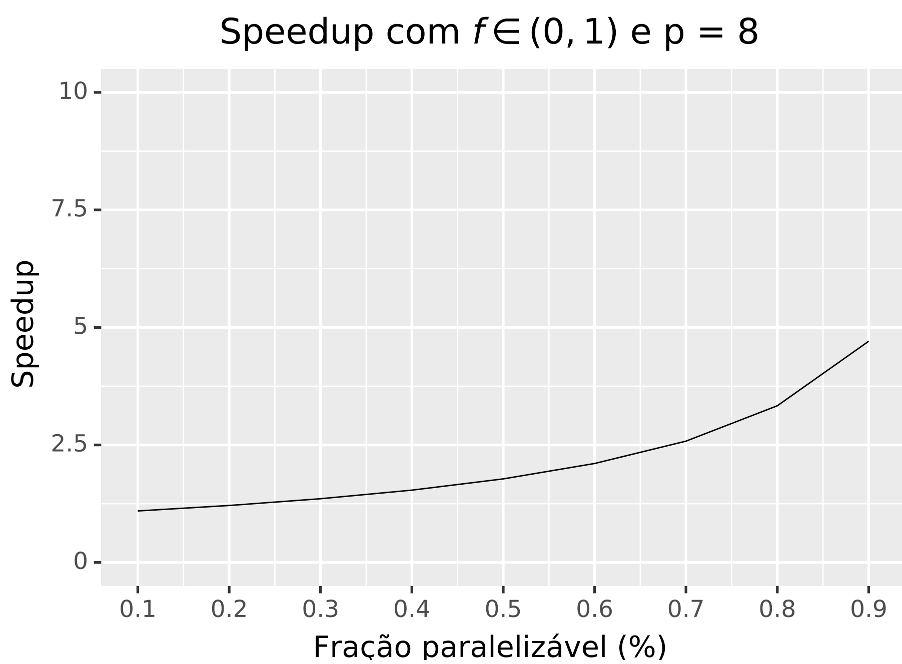
3) Plote agora o speedup em função da fração paralelizável (f) de um programa, para o intervalo $0 < f < 1$, supondo um sistema com:

- (a) 8 processadores
- (b) 128 processadores

Este exercício apresenta um outro ponto de vista do que foi apresentado no exercício anterior. Faz isto através da variação do fator de paralelização (f), de modo que a influência da influência da fração paralelizável f de um programa seja visulizada.

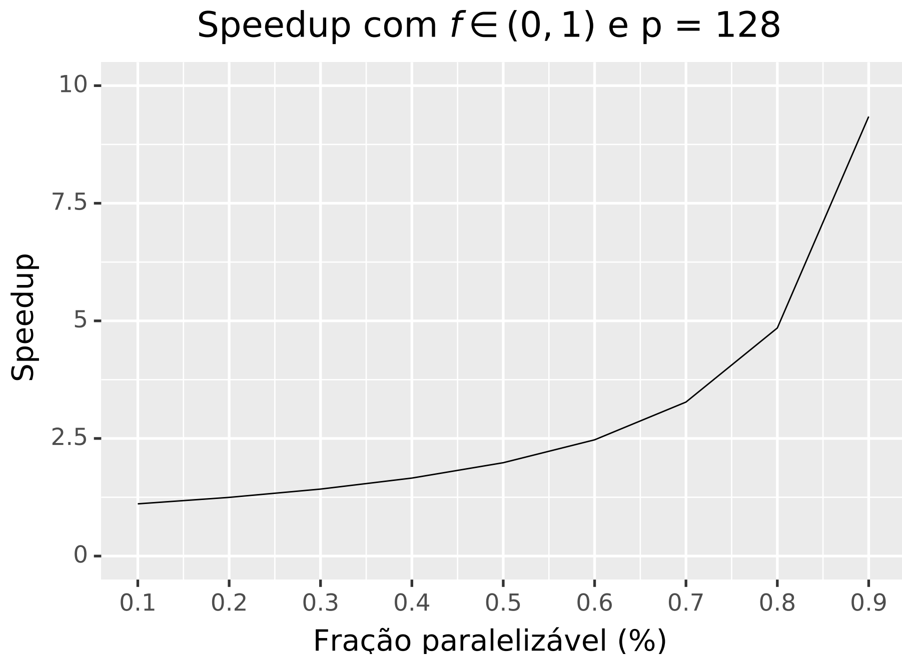
- (a) 8 processadores

```
## <ggplot: (8754884151153)>
```



- (b) 128 processadores

```
## <ggplot: (8754884221785)>
```



Ao analisar as figuras é possível perceber a influência da fração paralelizável do programa. Ao comparar os gráficos, até certo ponto, os dois sistemas, mesmo possuindo uma quantidade muito diferente de processadores, acabam tendo resultados próximos, mesmo havendo diferenças significativas na quantidade de recursos de cada um dos sistemas analisados. Assim, é possível perceber que, como apresentado em sala de aula, se o programa não consumir todos os recursos computacionais de modo a tirar o máximo proveito do mesmo, tem-se disperdício de recursos, financeiros e computacionais.

4) Considere um programa no qual a fração paralelizável corresponde a 90% do tempo de uma execução convencional em um processador.

- (a) Calcule o speedup que seria obtido num sistema com 16 processadores.

$$S_P = \frac{1}{1 - f + \frac{f}{P}} = \frac{1}{1 - 0.9 + \frac{0.9}{16}} = 6.4$$

- (b) Calcule o speedup que seria obtido num sistema com 64 processadores.

$$S_P = \frac{1}{1 - f + \frac{f}{P}} = \frac{1}{1 - 0.9 + \frac{0.9}{64}} = 8.767123$$

- (c) Quantas vezes o sistema com 64 processadores é mais rápido que o sistema com 16 processadores para este programa?

Esta relação pode ser determinada como a equação apresentada abaixo

$$\frac{8.767123}{6.4} = 1.369863$$

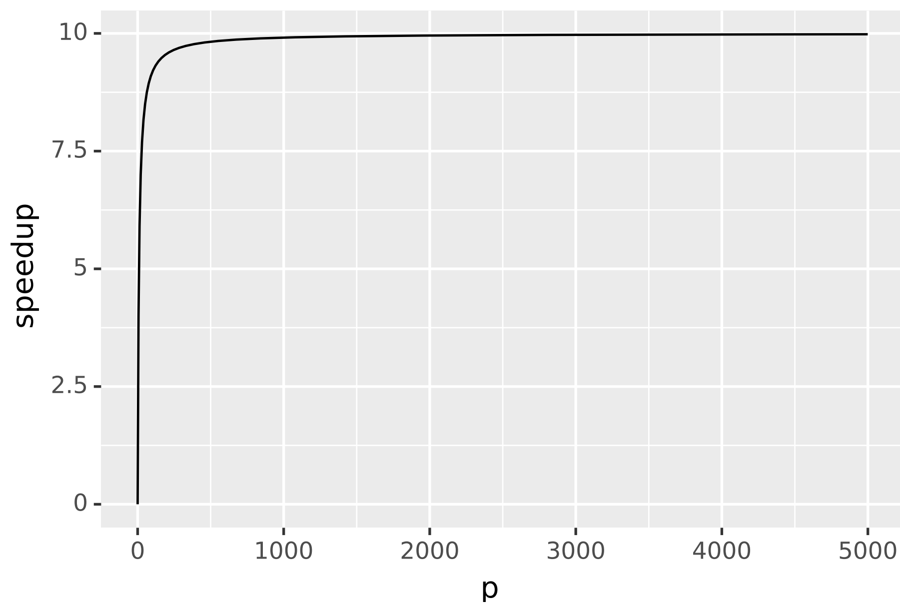
- (d) Quantos processadores são necessários para executar este programa na metade do tempo da execução no sistema com 16 processadores? (Justifique a resposta)

Para este exercício, antes de realizar testes aplicando a lei de Amdahl para verificar a quantidade de processadores necessários, o que computacionalmente poderia ser realizado de forma simples, façamos a análise do `Speedup máximo` que este programa pode alcançar.

$$\frac{1}{1-f} = \frac{1}{1-0.90} \approx 10$$

Ao verificar que o speedup máximo, para este contexto, é aproximadamente 10, fica claro que não é possível melhorar a execução a ponto de diminuir sua execução pela metadade do que foi alcançado com 16 processadores. Para confirmar tal questão, a visualização abaixo apresenta uma extrapolação dos valores de P considerando sua variação em um intervalo $0 \leq P \leq 5000$

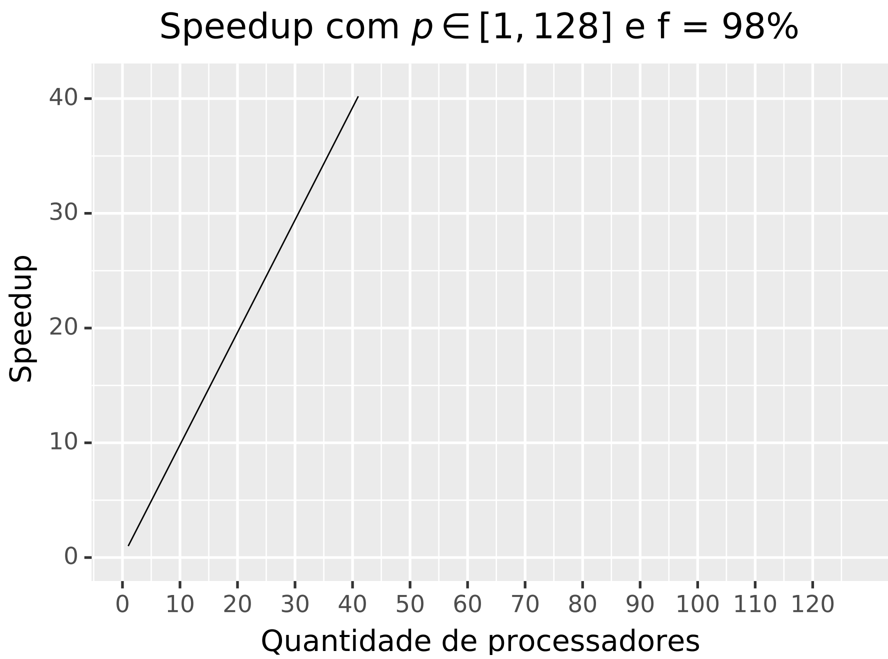
```
## <ggplot: (8754884132541)>
```



5) Utilizando as mesmas escalas dos gráficos construídos acima em `2-a` e em `3-a`, respectivamente, plote o `speedup-em-escala` para os seguintes casos:

- (a) $f = 98\%, 1 \leq P \leq 128$

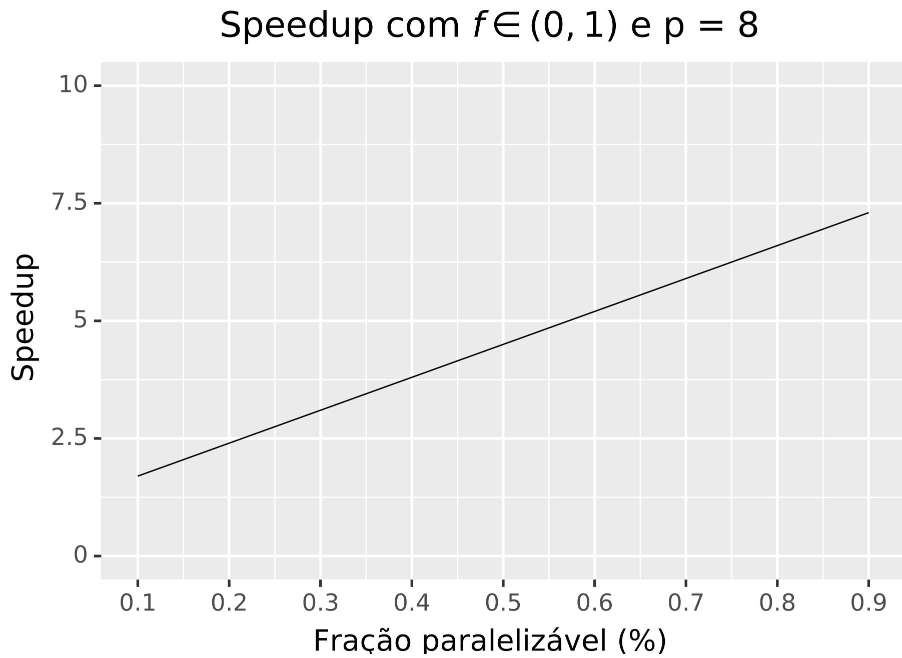
```
## <ggplot: (8754885347277)>
```



Comparando com o gráfico do exercício 2a, que possuí os mesmos valores, é possível perceber que ao realizar a consideração do speedup em escala, tem-se um ganho que cresce muito mais rápido que o speedup de Amdahl. Isto ocorre por conta do ponto de vista que está sendo considerado neste calculo, onde problemas maiores serão aplicados no sistema em questão.

- (b) 8 processadores, $0 < f < 1$

```
## <ggplot: (8754884059049)>
```


Speedup com $f \in (0, 1)$ e p = 8
10
7.5
5
2.5
0
Speedup
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Fração paralelizável (%)