Primeira Lista de Exercícios de 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 pedidos gráficos, 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 ?

  O pipeline trabalha utilizando um ciclo de *clock* único, de modo que o tamanho deste ciclo seja capaz de suportar a operação mais lenta (Patterson 2014), desta forma, o tempo de ciclo mínimo para este caso será $30s$.

- (b) Supondo que existam 100 tarefas a executar, qual o speedup em relação à execução num modo estritamente serial ?

  Primeiro determinamos o valor em relação a execução sem pipeline.

$$SemPipeline = N * K = 100 * (15 + 25 + 30 + 20) = 9000$$

  Agora, façamos a consideração do modo de execução com pipeline

$$ComPipeline = K + (N - 1) = 4 + (100 - 1) = 103$$

  Nesta parte, é considerado que o valor $SemPipeline$ está sendo `quantificado` em segundos e que o $ComPipeline$ em ciclos, desta forma, para tornar comparável, faz-se a multiplicação do $ComPipeline$ por 30s, já que este representa o tempo de cíclo único utilizado no pipeline.

$$ComPipeline_S = ComPipeline * 30 = 3090$$

  Por fim, o Speedup pode ser calculado

$$Speedup = \frac{N * K}{K + (N - 1)} = \frac{SemPipeline}{ComPipeline_S} = \frac{9000}{3090} = 2.912621$$

Com isto, é possível afirmar que o Speedup entre um modo pipeline e um estritamente serial é 2.912621.

- (c) Caso seja possível subdividir uma das 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 ?

Para reduzir a necessidade de um ciclo mínimo grande, a sub-tarefa escolhida para a subdivisão será a com tempo de execução igual a $30s$

- (d) Após a divisão proposta no ítem anterior, qual o novo speedup possível em relação à execução estritamente serial das 100 tarefas ?

Inicialmente é necessário definir qual será o novo tamanho do ciclo, uma vez que o maior ciclo que antes era $30s$ deixou de existir, assim, o novo tamanho de ciclo para o pipeline passa a ser $25s$, por possuir o maior tempo de execução. Com esta mudança, os mesmos passos apresentados no exercício 1(b) podem ser realizados. Note que, o valor $SemPipeline$ permanece igual, já que mesmo dividindo as atividades, o valor total de tempo utilizado na conta será o mesmo.

$$ComPipeline = K + (N - 1) = 5 + (100 - 1) = 104$$

Agora o cálculo de tempo é aplicado, considerando o valor $25s$

$$ComPipeline_S = ComPipeline * 25 = 2600$$

Calculando o Speedup tem-se que

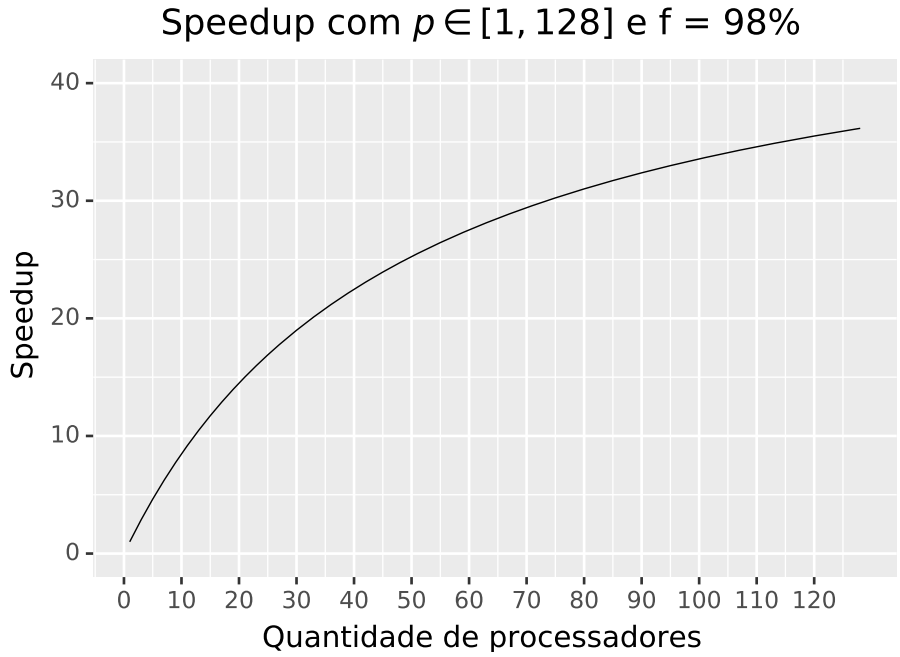
$$Speedup = \frac{N * K}{K + (N - 1)} = \frac{SemPipeline}{ComPipeline_S} = \frac{9000}{2600} = 3.461538$$

Com isto, com a mudança no tamanho das sub-tarefas, tem-se um Speedup de 3.461538 com relação ao modo estritamente serial.

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: (8772630145845)>
```

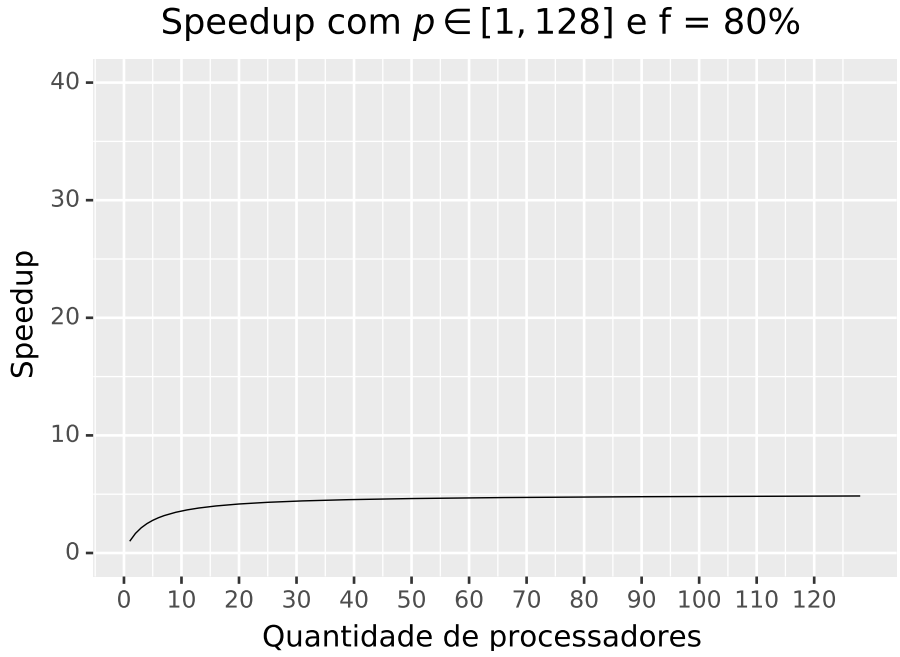


Ao considerar a plotagem realizada, é possível perceber a influência da quantidade de processadores no Speedup quando a fração de paralelização do programa analisado é alto, neste, quanto maior a quantidade de processadores, maior o Speedup.

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

- (b) 80%

```
## <ggplot: (8772626322485)>
```



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.

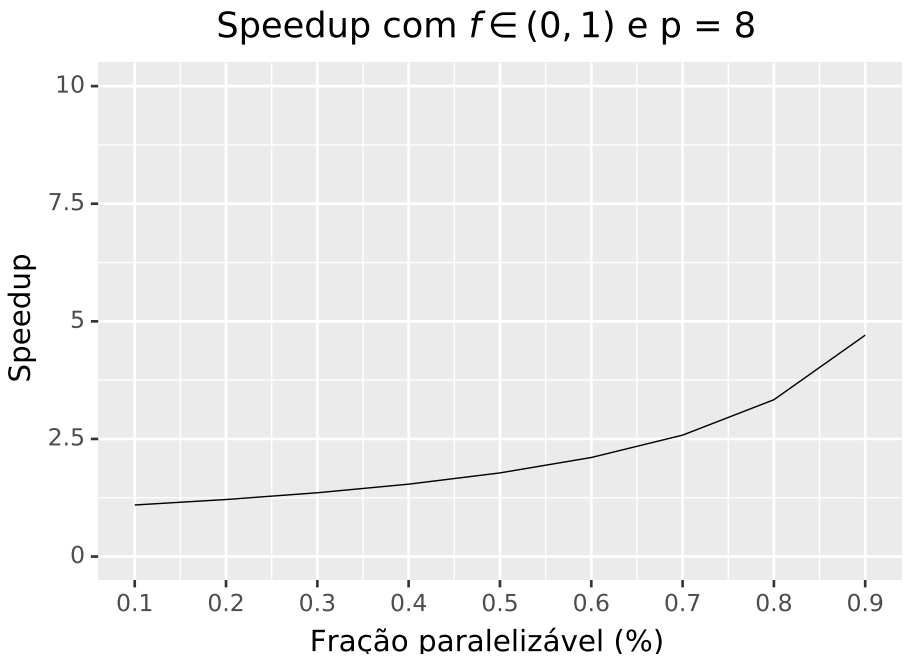
3) Plote agora o speedup em função da fração paralelizável (f) de um programa, para o intervalo $0 < f < 1$, supondo um sistema com:

- (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 fração paralelizável f de um programa seja visulizada.

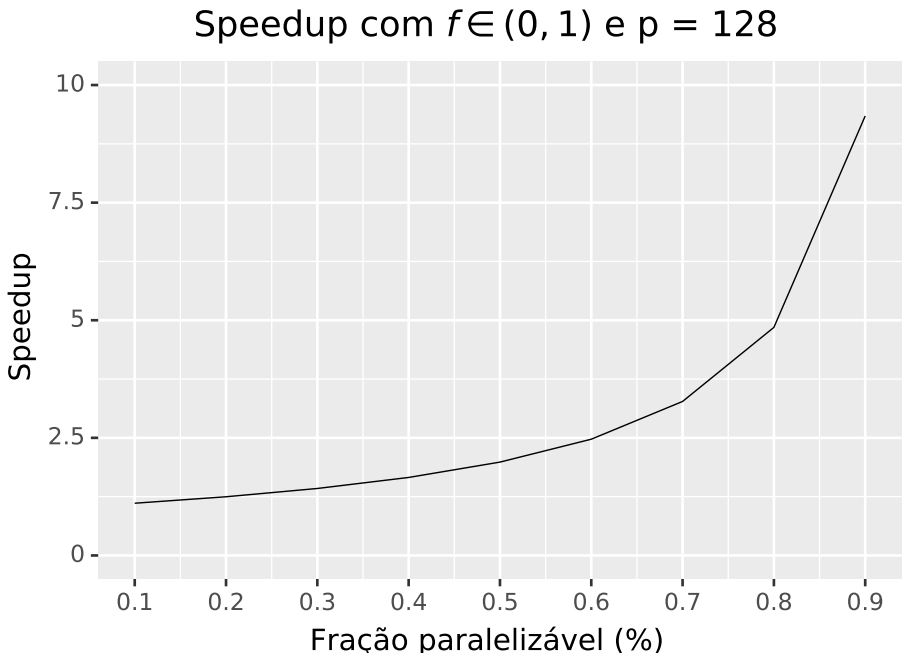
- (a) 8 processadores

```
## <ggplot: (8772626233121)>
```



- (b) 128 processadores

```
## <ggplot: (8772629310053)>
```



Ao analisar as figuras é possível perceber a influência da fração paralelizável do programa, veja que até certo ponto os dois sistemas acabam tendo resultados próximos, mesmo havendo diferenças significativas na quantidade de recursos de cada um. Assim, é possível perceber que, como apresentado em sala de aula, se o programa não consumir 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

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

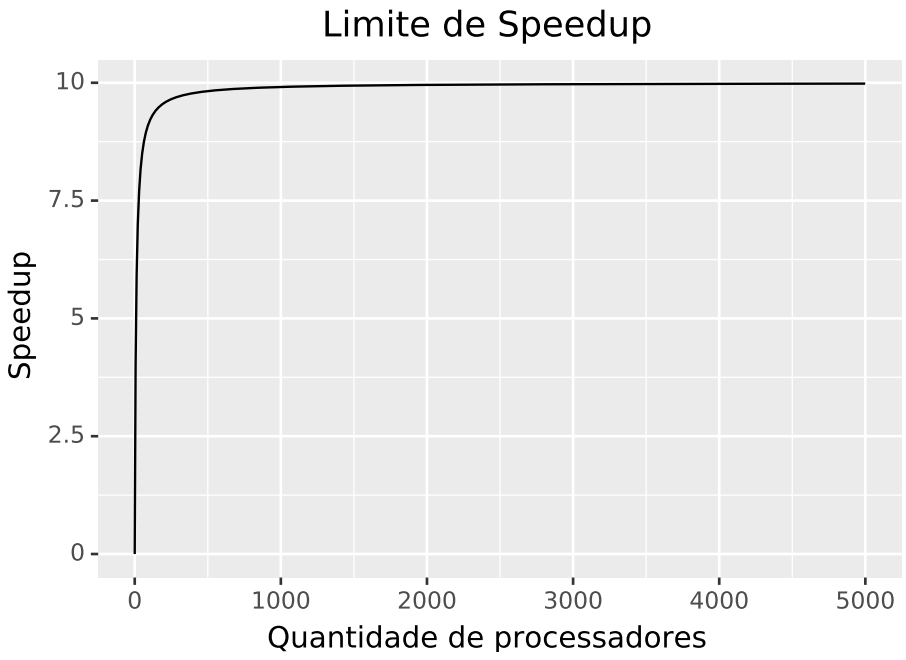
- (d) Quantos processadores são necessários para executar este programa na metade do tempo da execução no sistema com 16 processadores? (Justifique a resposta)

  Para este exercício, antes de realizar testes diretamente com o Speedup 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 $0 \leq P \leq 5000$

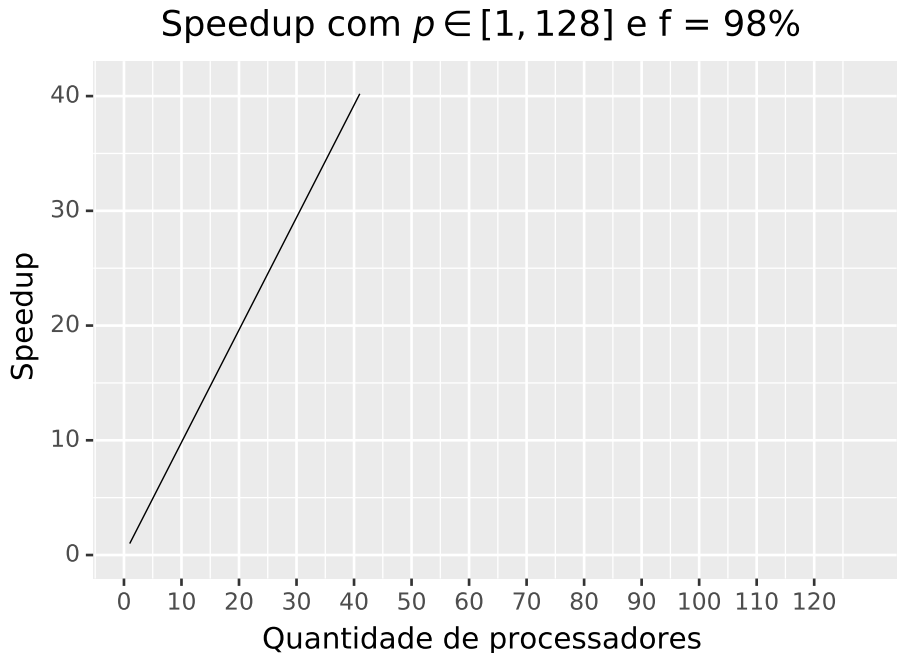
```
## <ggplot: (8772626218305)>
```



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:

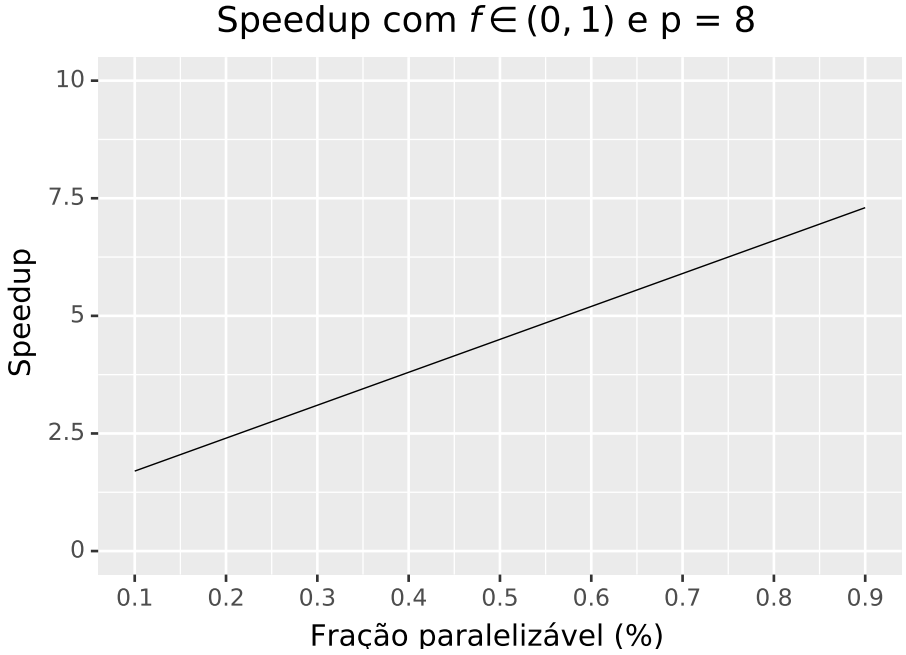
- (a) $f = 98\%, 1 \leq P \leq 128$

```
## <ggplot: (8772626205001)>
```



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

```
## <ggplot: (8772626137929)>
```



Ao comprar os resultados apresentados com a utilização do `speedup-em-escala` com os resultados apresentados nos exercícios 2-a e 3-a, é possível perceber que neste speedup, por considerar um cenário diferente, onde a quantidade de recursos dita a dimensão do problema que está sendo trabalho, tem-se resultados com crescimento de speedup muito maior.

Referências

Patterson, David. 2014. *Organização E Projeto de Computadores : Interface Hardware/Software*. Rio de Janeiro, Brasil: Elsevier.