Kyle Tranfaglia

COSC 420

Exercise 03

15 Oct. 2024

Performance Analysis with Amdahl's Law

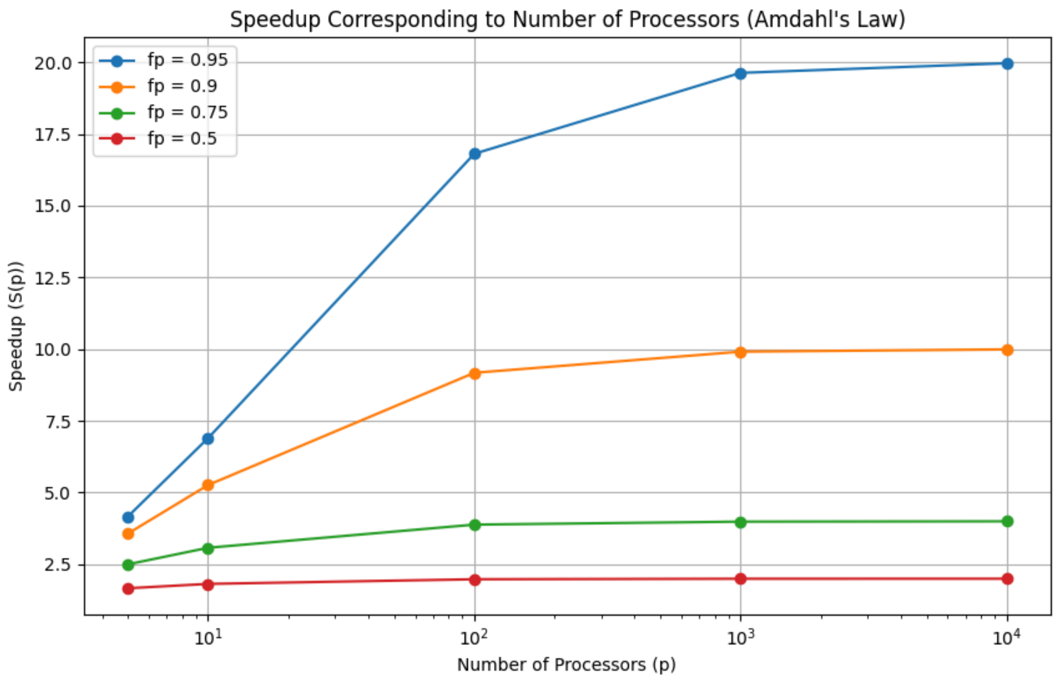
The following was produced using a Python program utilizing matplotlib.pyplot:

Speedup values for fp = 0.95:
S(5) = 4.1667
S(10) = 6.8966
S(100) = 16.8067
S(1000) = 19.6271
S(10000) = 19.9621

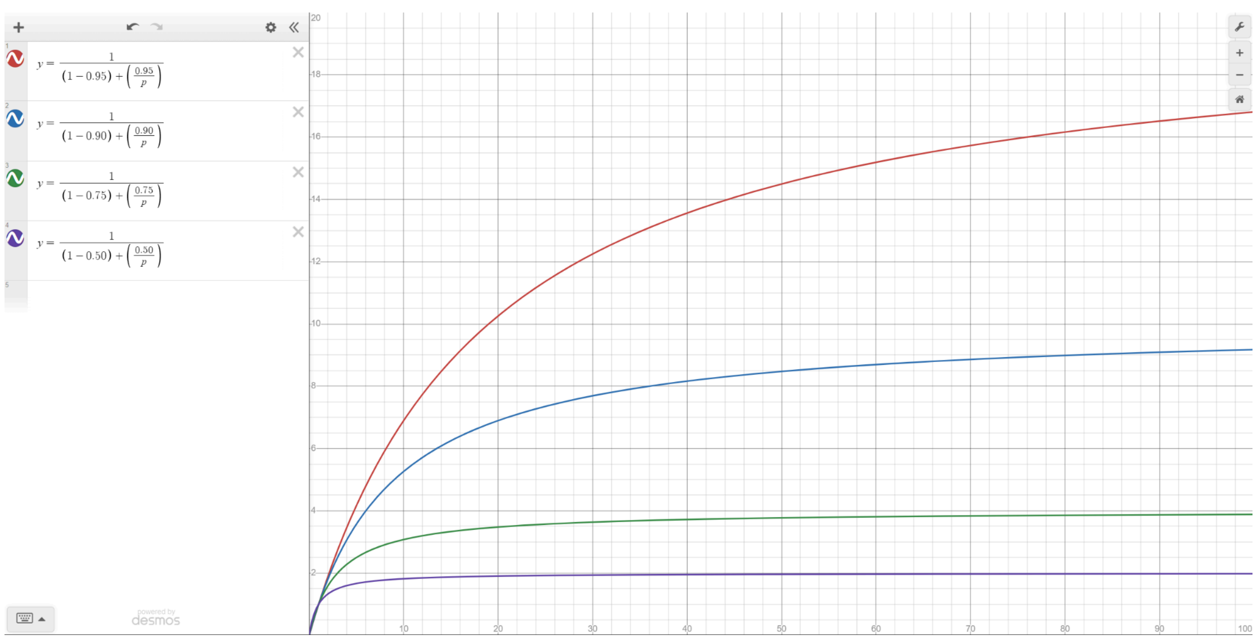
Speedup values for fp = 0.9:
S(5) = 3.5714
S(10) = 5.2632
S(100) = 9.1743
S(1000) = 9.9108
S(10000) = 9.9910

Speedup values for fp = 0.75:
S(5) = 2.5000
S(10) = 3.0769
S(100) = 3.8835
S(1000) = 3.9880
S(10000) = 3.9988

Speedup values for fp = 0.5:
S(5) = 1.6667
S(10) = 1.8182
S(100) = 1.9802
S(1000) = 1.9980
S(10000) = 1.9998



The following secondary line graph was constructed using Demos Online Graphing Calculator:



Performance analysis given the following output:

Serial Program:

summation from 1 to 1000000000: 500000000500000000

runtime: 0.98

2 core program:

MPI runtime Rank 1: 0.5

Summation from 1 to 1000000000: 500000000500000000

MPI runtime Rank 0: 0.5

8 Core program:

MPI runtime Rank 1: 0.23

MPI runtime Rank 7: 0.23

MPI runtime Rank 4: 0.24

MPI runtime Rank 6: 0.24

MPI runtime Rank 5: 0.24

MPI runtime Rank 3: 0.24

MPI runtime Rank 2: 0.25

Summation from 1 to 1000000000: 500000000500000000

MPI runtime Rank 0: 0.25

We can extract the following from the data:

Serial runtime: Tserial = 0.98

2-core parallel runtime: Average Tparallel = (0.5 + 0.5) / 2 = 0.5

8-core parallel runtime: Average Tparallel = (0.23 + 0.23 + 0.24 + 0.24 + 0.24 + 0.24 + 0.25 + 0.25) / 8 = 0.24

(a)

Since speedup is the ratio of the serial runtime to the parallel runtime, we can use

S(p) = (Tserial / Tparallel)

Tserial is the runtime of the serial program, and Tparallel is the parallel runtime.

So,

S(2) = Tserial / Tparallel = 0.5 / 0.98 = 1.96

S(8) = Tserial / Tparallel = 0.98 / 0.24 ≈ 4.08

(b)

Since efficiency is the ratio of speedup to the number of processors, which measures how effectively the processors are used, we can use

E(p) = S(p) / p

So,

E(2) = S(2) / 2 = 1.96 / 2 = 0.98 >> 98%

E(8) = S(8) / 8 = 4.08 / 8 ≈ 0.51 >> 51%

(c)

Since the parallel fraction is the portion of the program that can be parallelized and calculated using Amdahl's Law, we can use

Fp = (S(p) - 1) / (S(p) • (1 - (1 / p))

So,

(S(2) - 1) / (S(2) • (1 - (1 / 2)) = (1.96 - 1) / (1.96 • 0.5) = 0.96 / 0.98 ≈ 0.98 >> 98%

(S(8) - 1) / (S(8) • (1 - (1 / 8)) = (4.08 - 1) / (4.08 • 0.875) = 03.08 / 3.57 ≈ 0.86 >> 86%

(d)

Finally, since we know that Fp + Fs = 1 and that Fs is the fraction of the program that is serial and cannot be parallelized, then

Fs = 1 - Fp

So,

Fs(2) = 1 - Fp(2) = 1 - 0.98 = 0.2 >> 2%

Fs(8) = 1 - Fp(8) = 1 - 0.86 = 0.14 >> 14%

In summary, we computed the following results:

Speedup (2 cores): 1.96

Speedup (8 cores): 4.08

Efficiency (2 cores): 98%

Efficiency (8 cores): 51%

Parallel Fraction (2 cores): 98%

Parallel Fraction (8 cores): 86%

Serial Fraction (2 cores): 2%

Serial Fraction (8 cores): 14%