Speedup Models

3 speedup performance models

- Fixed Load Speedup (Amdahl's Law-1967)
- Fixed Time Speedup (Gustafson's law-1987)
- Memory-Bounded Speedup (Sun and Ni-1993)

Amdhal's Law (1967)

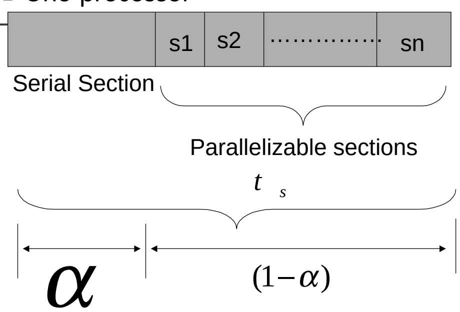
- Assumption: computational load is fixed with a fixed problem size
- If we have more processors, the fixed load is distributed to more processors
- Objective to get fast results.

Amdhal's Law

- Amdahl's law implies that the sequential portion of the program does not change with respect to machine (processor) size
- However, parallel portion is evenly executed by P processors, resulting in reduced time

Amdahl’s Law(Fixed Problem Size)

- One processor

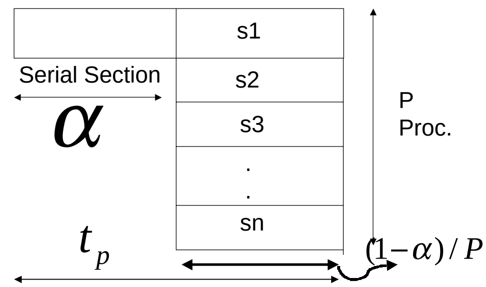


α: PERCENTAGE OF A PROGRAM EXECUTED SEQUENTIALLY

Amdahl’s law

(Parallelizing sequential Problem)

- Multiple Processors



Amdahl’s law

- Speedup:

  S(n) = T(sequential)/T(processor)

$$\frac{\alpha+(1-\alpha)}{\alpha+\frac{(1-\alpha)}{P}} = \frac{1}{\alpha+(1-\alpha)/P} = \frac{P}{1+(P-1)\alpha}$$

$$\frac{P}{1+(P-1)\alpha} = \frac{1}{\alpha+(1-\alpha)/P}$$

$$\alpha = 0, S = P$$

$$\alpha = 1, S = 1$$

$$P \xrightarrow{\infty} S = \frac{1}{\alpha}$$

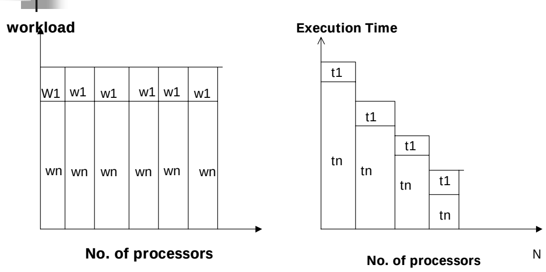
Speedup is upper bounded by $\frac{1}{\alpha}$

Amdhal’s Law

- As α increases, speedup decreases
- With a small percentage of sequential code, entire performance can’t go higher than α
- **Sequential Bottleneck** α : Can’t be solved by increasing number of processors. Problem lies in sequential portion of code

Amdhal’s Law

- As the number of processors increases, load on each processor decreases
- However total amount of work is kept constant
- Execution time decreases
- Eventually sequential part will dominate the performance as P becomes very large



Amdhal’s Law

- Fixed load prevents scalability in performance
- Impact on Parallel Computer industry:
  - Manufacturers were discouraged from making large-scale parallel computers
  - More research attention was shifted toward developing parallelizing compilers which will reduce the value of α
  - Boost performance

Fixed Time Speedup

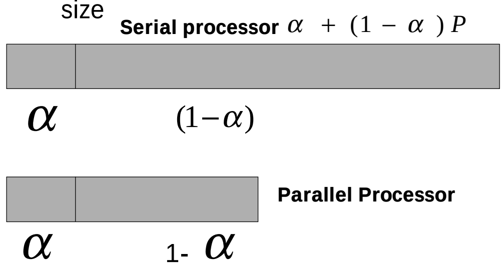
- Problem cannot scale to match available computing power as machine size increases in Amdahl's law
- α is a bottleneck but can be alleviated by removing fixed load

Fixed Time Speedup for Scaled Problems

- Assumption: Largest problem will require largest machine; smallest problem will require smallest machine→but both should have the same run time.
- So, same amount of parallel portion is executed on every processors since problem size increases

Gustafson's Law (1988)

- Increase Problem Size with machine size



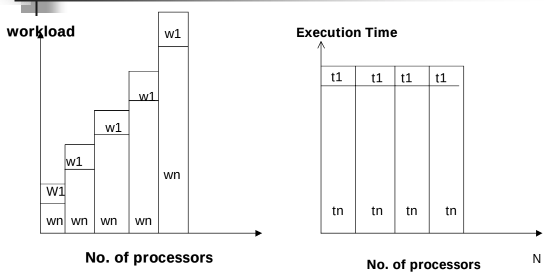
Gustafson's Law

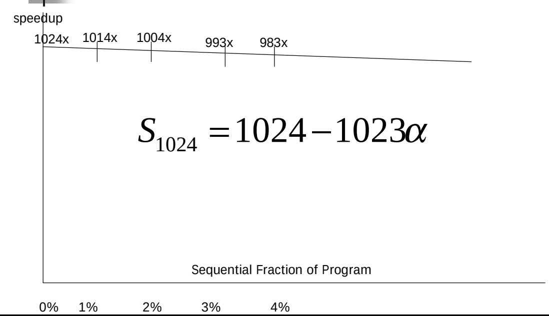
- Scaled Speedup =

$$\frac{(\alpha + (1-\alpha)P)}{\alpha + (1-\alpha)} = \alpha + (1-\alpha)P$$

$$= P - \alpha(P-1)$$

- Claim: Serial code (S) does not increase as the problem size $S + \beta P$





$$S_{1024} = 1024 - 1023\alpha$$

Gustafson's law

- Gustafson'e law supports scalable performance as machine size increases
- It keeps all processors busy by increasing problem size
- Sequential portion no longer a bottleneck

Scalability

- Architecture/Hardware Scalability:
  - Increase in size to increase in performance
- Algorithmic Sclability:
  - Parallel algorithm accommodates increased data items with a low and bounded increase in computational steps