HIGH PERFORMANCE COMPUTING for SCIENCE & ENGINEERING (HPCSE) I

HS 2022

EXERCISE 01: AMDAHL'S LAW, ROOFLINE MODEL, CACHE

Noah Baumann

Computational **S**cience and **E**ngineering **Lab**
ETH Zürich

28.09.2022

Outline

I. Amdahl’s Law - Question 1
II. Parallel scaling - Question 2
III. Roofline model - Question 3
IV. Cache size& speed – Question 4 & 5

I. Amdahl's law

T : total execution time of program

p : parallel percentage/ fraction of program

n : number of processors to run parallel fraction

1 processor $T(1) = (1-p)T + pT = T$

n processors $T(n) = (1-p)T + \frac{pT}{n}$

Speed-up $$S(n) = \frac{T(1)}{T(n)} = \frac{(1-p)T + pT}{(1-p)T + \frac{p}{n}T} = \frac{1}{1-p+\frac{p}{n}}$$

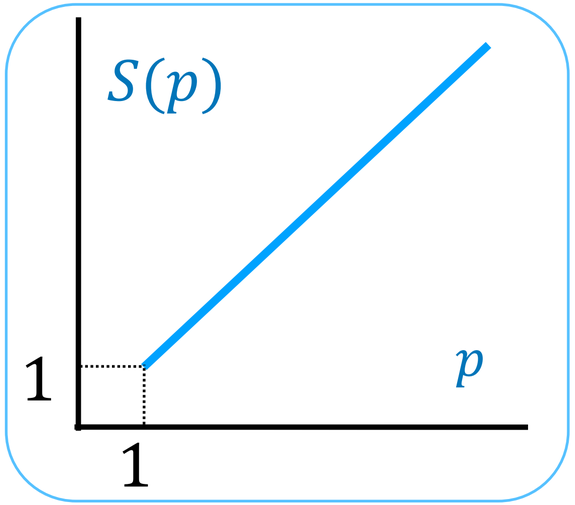
Questions 1a), 1b) & 1c): Use Amdahl's law formula

1d): Search for $T(n) = T_{serial} + T_{communicate}(n) + T_{parallel}(n)$

II. Parallel scaling

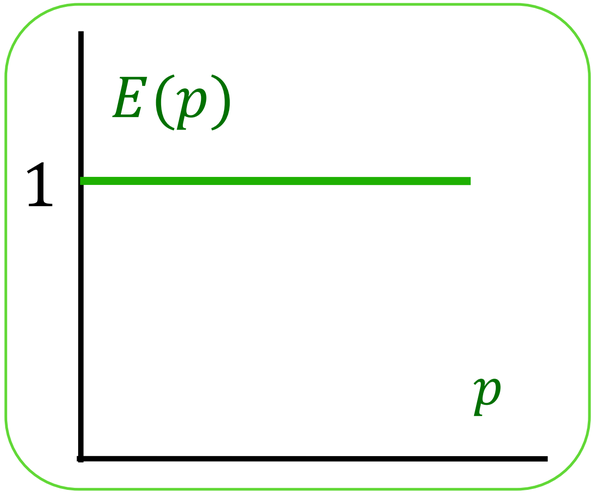
a. Strong scaling : Speed-up on n processors : $S(p) = T(1)$ $/T(p)$

- Perfect strong scaling: linear $S(p)$ vs. p



b. Weak scaling: Efficiency on n processors $E(p) = T(1)$ $/T(p)$ (speed-up with proportional increase of problem size)

- Perfect weak scaling: constant line $E(p) = 1$

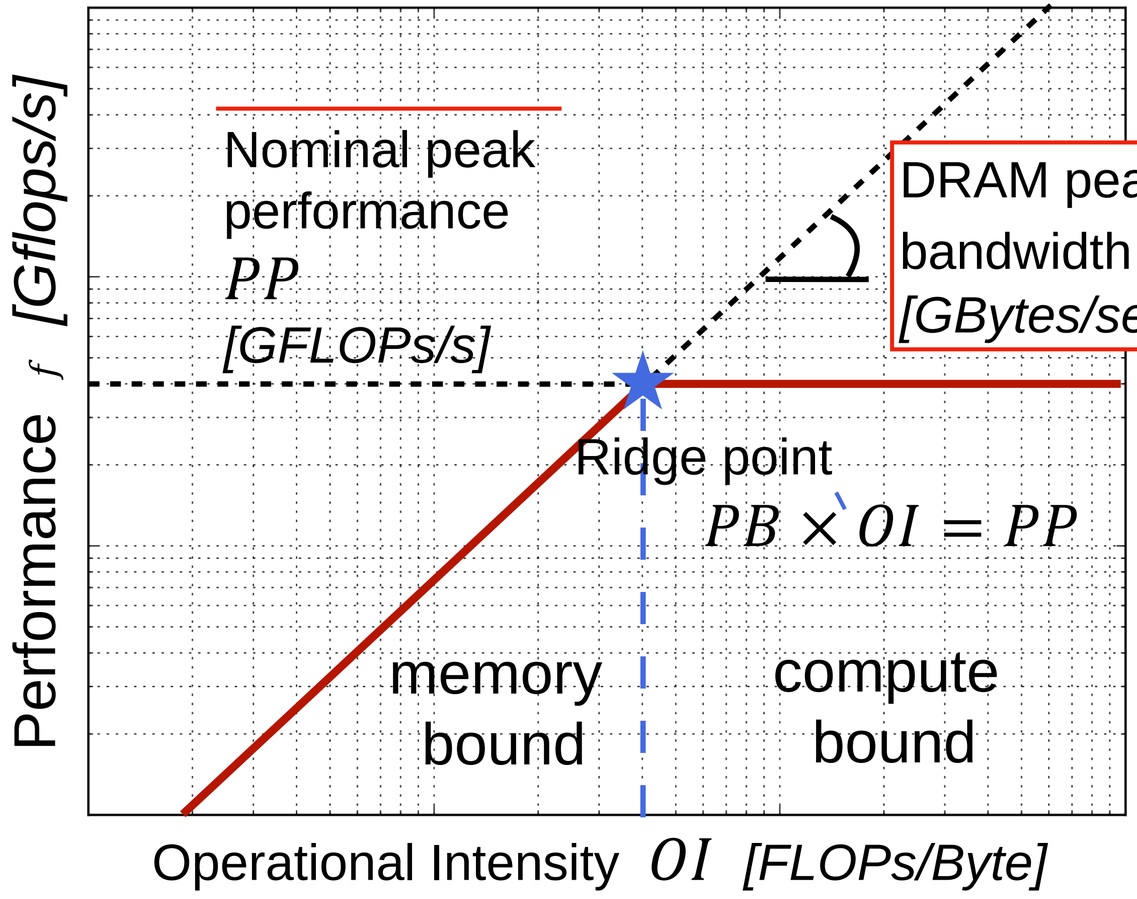


1. Strong scaling plot
2. Weak scaling plot

II. Roofline Model

Log-log plot

$$f = min(PB \times OI, PP)$$



PP, PB : from hardware

OI : from algorithm/ kernel

HOW TO Calculate PP, PB from hardware.

Processor frequency

Peak Processor FLOPs

$$PP = \left[\frac{cycles}{s}\right] \times \left[\frac{FLOPs}{cycle}\right] \times [\#\ cores]$$

Memory frequency/ speed

Memory channels

$$PB = \frac{\left[\frac{cycles}{s}\right] \times [\#\ channels] \times [channel\ size\ [bits]]}{[bits/Byte] = 8}$$

II. Roofline Model

$$PP = \left[\frac{cycles}{s}\right] \times \left[\frac{FLOPs}{cycle}\right] \times [\#\ cores]$$

Example 1: Euler Intel Xeon Gold 6150

frequency = 2.7 GHz

Intel AVX-512: 512 bit —>
16x single precision (32 bit) or 8x double precision (64 bit) registers
2x16 single FLOP/cycle or 2x8 double FLOP/cycle x FMA units

$$\text{—>}\ 2 \times 2 \times 16 = 64 \left[\frac{FLOPs}{cycle}\right]$$

1 node = 18 cores

$$PP_1 = 2.7 \left[G\frac{cycles}{s}\right] \times 64 \left[\frac{FLOP}{cycle}\right] \times 18[\#\ cores]$$

$$= 3110 \left[\frac{GFLOP}{s}\right]$$

$$= 172.8 \left[\frac{GFLOP}{s}\right] / core$$

Example 2: Intel Core i7-7660U (my Laptop)

frequency = 2.5 GHz

Intel AVX2: 256 bit —>
8x single precision (32 bit) or 4x double precision (64 bit) registers
2x8 single FLOP/cycle or 2x4 double FLOP/cycle x FMA units

$$\text{—>}\ 2 \times 2 \times 8 = 32 \left[\frac{FLOPs}{cycle}\right]$$

2 cores

$$PP_1 = 2.5 \left[G\frac{cycles}{s}\right] \times 32 \left[\frac{FLOP}{cycle}\right] \times 2[\#\ cores]$$

$$= 160 \left[\frac{GFLOP}{s}\right] \qquad = 80 \left[\frac{GFLOP}{s}\right] / core$$

INFO FOR FLOPS/ ARCHITECTURE https://en.wikichip.org/wiki/flops

II. Roofline Model

$$PB = \frac{\left[\frac{cycles}{s}\right] \times [\#\ channels] \times [channel\ size\ [bits]]}{[bits/Byte]}$$

Example 1: Euler Intel Xeon Gold 6150

https://en.wikichip.org/wiki/intel/xeon_gold/6150

memory frequency = 2.67 GHz

max 6 memory channels

channel size = 64 bits

$$PB_1 = \frac{2.67[GHz] \times 6 \times 64[bits]}{8[bits/B]}$$

$$= 128.2\left[\frac{GB}{s}\right]$$

Example 2: Intel Core i7-7660U (my Laptop)

memory frequency = 2.13 GHz

max 2 memory channels

channel size = 64 bits

$$PB_1 = \frac{2.13[GHz] \times 2 \times 64[bits]}{8[bits/B]}$$

$$= 34.1\left[\frac{GB}{s}\right]$$

INFO FOR FLOPS/ ARCHITECTURE https://en.wikichip.org/wiki/flops

II. Roofline Model

Example 1: Euler Intel Xeon Gold 6150

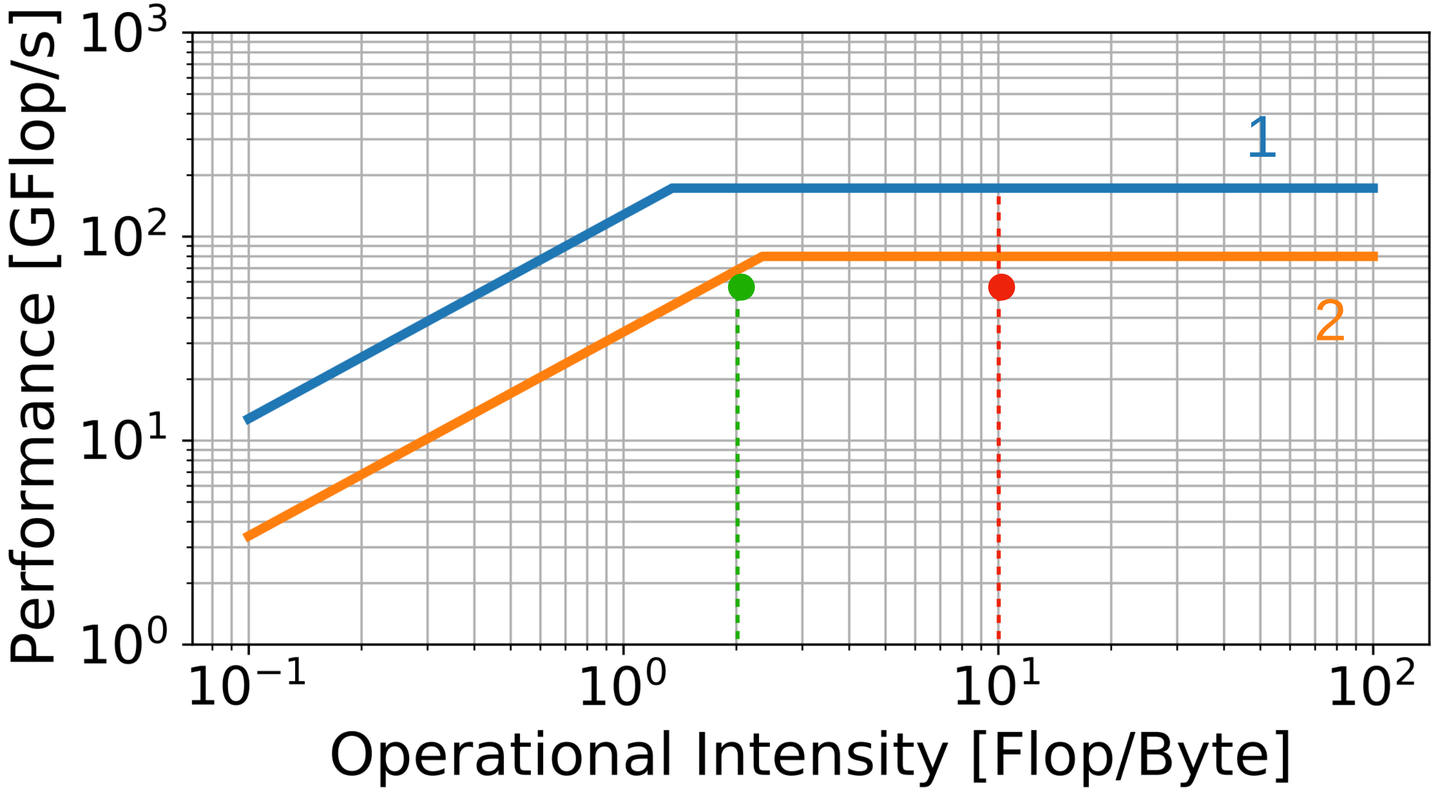
$$PP_1 = 172.8 \left[\frac{GFLOPs}{s}\right]/core \quad PB_1 = 128.2 \left[\frac{GB}{s}\right]$$

$$\rightarrow OI_1^* = \frac{172.8}{128.2} = 1.34 \frac{FLOPs}{B}$$

Example 2: Intel i7-7660U

$$PP_2 = 80 \left[\frac{GFLOPs}{s}\right]/core \quad PB_2 = 34.1 \left[\frac{GB}{s}\right]$$

$$\rightarrow OI_2^* = \frac{80}{34.1} = 2.3 \frac{FLOPs}{B}$$



- If my kernel has $OI = 2 \frac{FLOPs}{B}$,
- If my kernel has $OI = 10 \frac{FLOPs}{B}$,

Which kernel is compute bound for which cpu?
Which kernel performs relatively better to each other?

II. Roofline Model

HOW TO compute Operational Intensity from a given Kernel?

$$OI = \frac{W}{Q}\,[FLOP/byte]$$

W = amount of work / i.e floating point operations required

Q = memory transfer / i.e access from DRAM to lowest level cache

Example 1

```
float in[N], out[N];
for (int i=1; i<N-1; i++)
      out[i] = in[i-1]-2*in[i]+in[i+1]
```

float=4 byte, double=8 byte

A. Amount of flops W

For every `i`: `out[I] = in[i-1]-2*in[i]+in[i+1]` 3 flop

Loop over: `for (int i=1; i<N-1; I++)->` (N-2) repetitions

Total = 3(N-2) FLOPs

B. Memory acceses Q — Depends on cache size!

`out[I] = in[i-1]-2*in[i]+in[i+1]`

	For every `i`	Total Q	Total [bytes]	OI [flop/B]
1. No cache (we read directly from slow memory) every data accessed is counted →	4	4(N-2)	4(N-2)x4	$\frac{3}{16}$
2. Perfect cache (infinite size cache) data is read & written ONLY ONCE →	2	2(N-2)	2(N-2)x4	$\frac{3}{8}$

II. Roofline Model

How to compute Operational Intensity?

$$OI = \frac{W}{Q}\ [FLOP/byte]$$

Example 2 Matrix multiplication (Naive)

```
double A[N,N], B[N,N], C[N,N];
for (int j=0; j<N; j++)
    for (int i=0; i<N; i++)
        for (int k=0; k<N; k++)
            C[i,j] = C[i,j] + A[i,k]*B[k,j]
```

= 1 MUL + 1 ADD

A. Amount of flops W ? For every `i,j:` **`C[i,j] = C[i,j] + A[i,k]*B[k,j]`** 2 FLOPs

Loop over N*N*N —> Total = $2N^3$ FLOPs

B. Memory acceses Q ? For every `i,j:` **`C[i,j] = C[i,j] + A[i,k]*B[k,j]`**

	For every `i,j`	Total Q	Total [bytes]	OI [flop/B]
1. Perfect cache (small N - fits in cache) →	4 (3 read+ 1 write)	$4N^2$ Lower bound for Q!	$4N^2$x8	$\frac{2N^3}{32N^2} = \mathcal{O}(N)$
2. More realistic cache →	For every `C[i,j]` element: – read a row of `A (N)` = 2N read – read a column of `B (N)` – `read & write 1 element C` = 2N + 2	$(2N+2)N^2$	$(2N+2)N^2$x8	$\frac{2N^3}{8(2N^3+2N^2)} \approx \frac{1}{4}$