HIGH PERFORMANCE COMPUTING for SCIENCE & ENGINEERING (HPCSE) I

HS 2021

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

Ermioni Papadopoulou

Computational Science and Engineering Lab
ETH Zürich

08.10.2021

Outline

I. Amdahl's Law - Question 1
II. Parallel scaling - Question 2
III. Roofline model - Question 3
    (Cache & Questions 4, 5 next week)
IV. Debugging tutorial

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 $\quad T = (1-p)T + pT$

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

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

1. How does Eq.★ change? - extra terms?
2. Maximum speed up with n * cores. Remember

$$\text{argmax}_n f(n) \rightarrow \frac{\partial f}{\partial n} = 0$$

$n \rightarrow \infty$

Consider 3 cases:
1. Perfect load balance
2. Load imbalance of 1.5 and 3x for 1 core

Question 1: Amdahl's law (30 points)

a) Suppose you have a program where 99.99% of the runtime is parallelizable. You want to run this program on a super computer that has up to $2'000'000$ cores.
What is the maximum speed up you can achieve if you had no limitations of processors n? What speed up can you achieve with 20, 200, 2000, $20'000$, $200'000$ and $2'000'000$ cores?

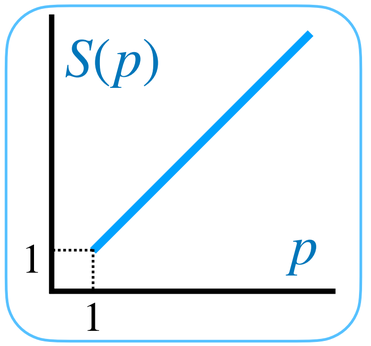
b) Not all parts of a program benefit from increasing the number of cores n used. Assume you have a communication operation that scales proportionally with the number of cores as $0.01n$. The serial fraction is 0.01.
What is the maximum speed up you can achieve? If you reduce the communication to $0.001n$ what is the new maximum speedup?

c) Upon inspection of a code, you see that 1000 of the code's operations are parallelizable and 10 are not. Suppose the time to compute one operation (both serial and parallel) is t_1.
- Compute the time $T(n)$ to execute the code with n cores.
- Using this time $T(n)$, compute the speed-up $S(n)$ for $n = 10$. Verify your answer using Amdahl's law.
- The speed up you get is true only in the case of perfect load balancing. For $n = 10$, re-compute the speed-up assuming one core is assigned 1.5 and 3 times more of the parallel operations.

II. Parallel scaling

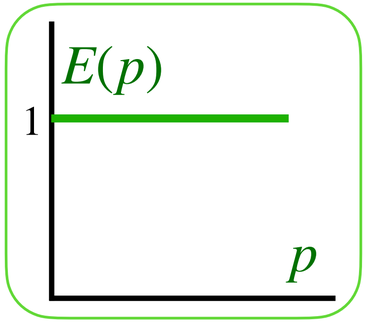
a. Strong scaling : Speed-up on n processors : $S(p) = T(1)/T(p)$

- 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 $E(p) = 1$



Question 2: OPTIONAL - Parallel Scaling (20 points)

A program simulates N particles, all particles interact with each other and, thus, the number of interactions is proportional to N^2. The runtimes of the program in seconds on P processor cores are in the table:

P\N	500	1000	1500	2000
1	6.00	30.00	72.00	120.00
4	1.50	7.50	18.00	30.00
9	0.75	3.50	9.00	20.00
16	0.50	2.15	6.00	12.00
24	0.40	1.50	4.50	10.00

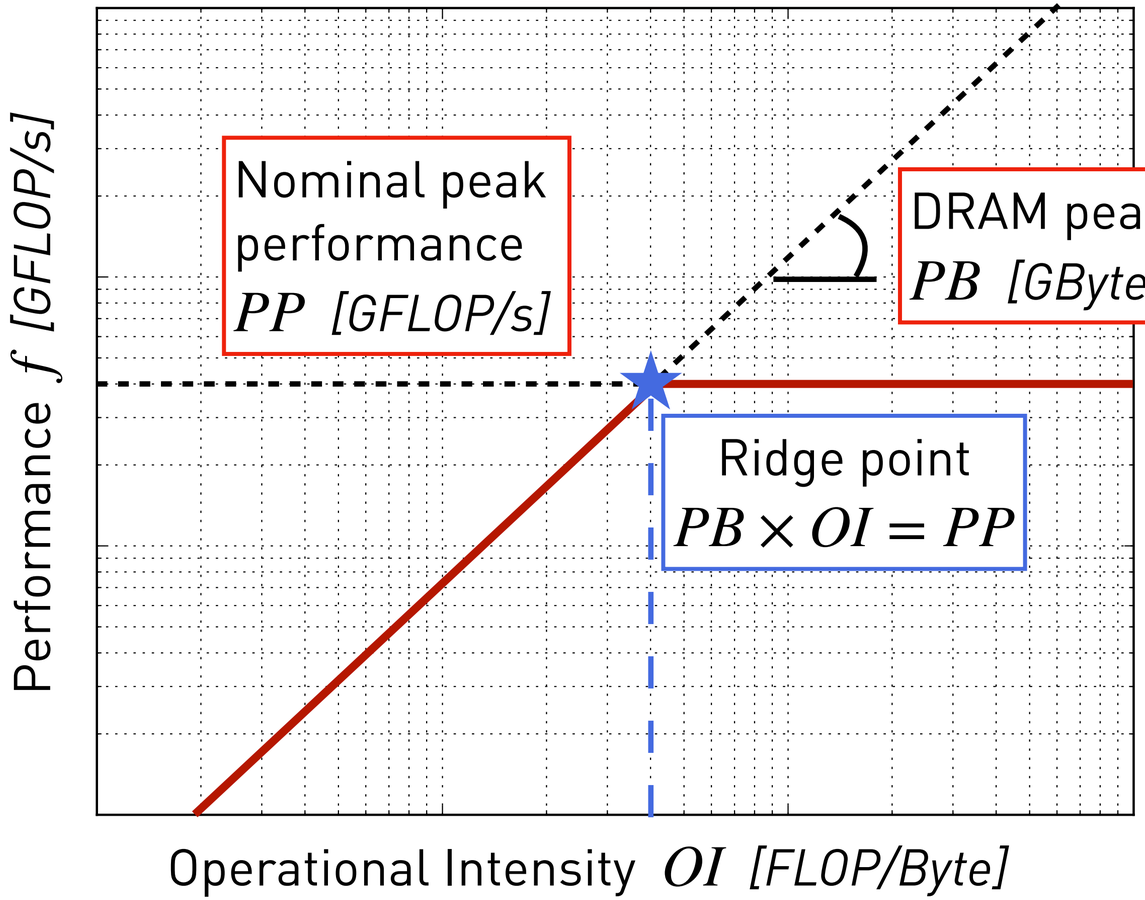
1. Strong scaling plot
2. Weak scaling plot

What is the problem size?

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

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

Memory frequency/ speed

Memory channels

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

II. Roofline Model

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

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 2 FMA units

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

1 node = 18 cores

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

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

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

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

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 2 FMA units

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

2 cores

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

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

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

II. Roofline Model

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

Example 1: Euler Intel Xeon Gold 6150

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

$\text{memory frequency} = 2.67\ \text{GHz}$

$\text{max 6 memory channels}$

$\text{channel size} = 64\ \text{bits}$

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

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

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

$\text{memory frequency} = 2.13\ \text{GHz}$

$\text{max 2 memory channels}$

$\text{channel size} = 64\ \text{bits}$

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

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

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

II. Roofline Model

Example 1: Euler Intel Xeon Gold 6150

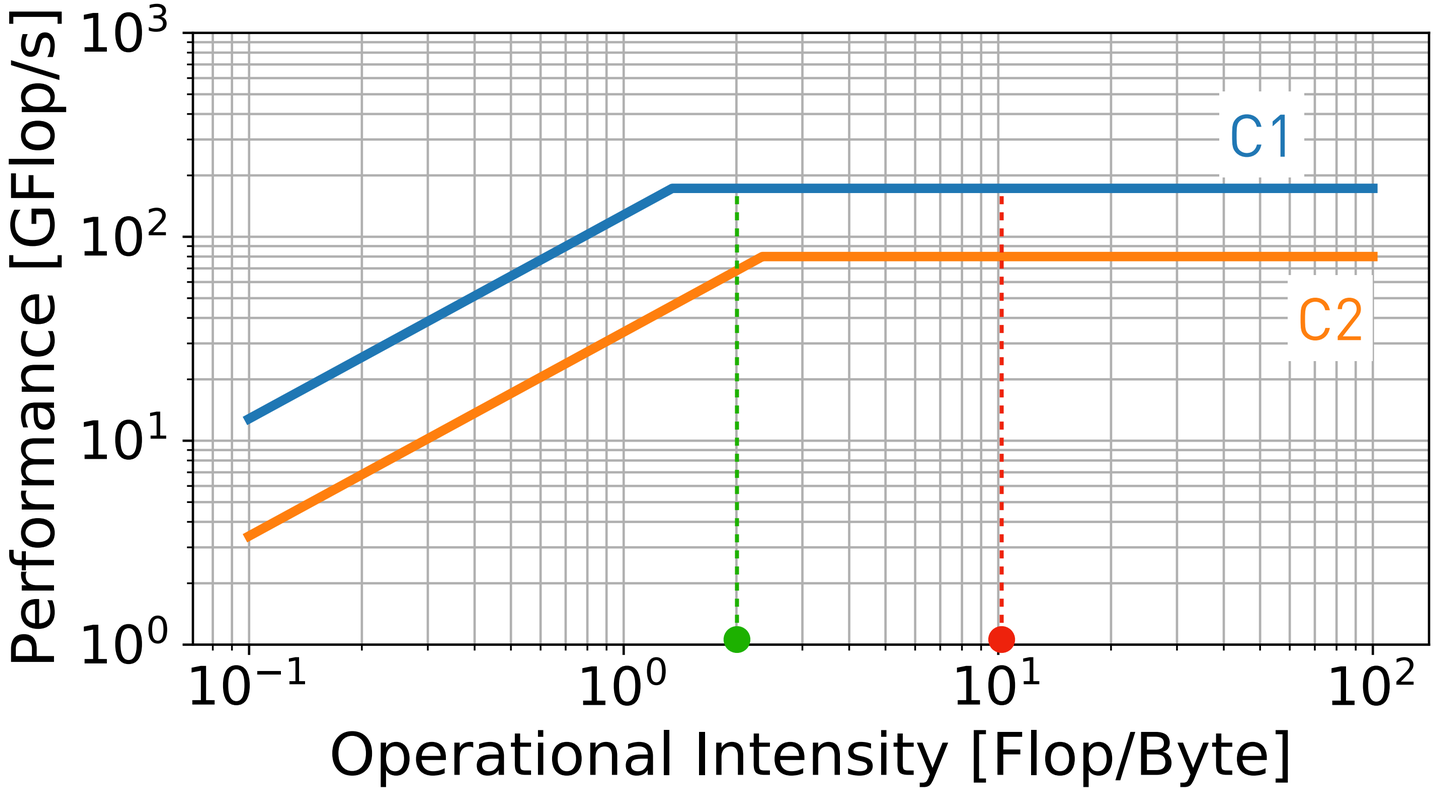
$$PP_1 = 172.8 \left[\frac{\text{GFLOP}}{s}\right]\text{/core} \quad PB_1 = 128.2 \left[\frac{\text{GB}}{s}\right]$$

$$\rightarrow OI_1^* = \frac{3110}{128.2} = 24.25 \ \frac{\text{FLOP}}{\text{B}}$$

Example 2: Intel i7-7660U

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

$$\rightarrow OI_2^* = \frac{160}{34.1} = 4.7 \ \frac{\text{FLOP}}{\text{B}}$$



- If my kernel has $OI = 2 \ \frac{FLOP}{B}$, which computer should I use?
- If my kernel has $OI = 10 \ \frac{FLOP}{B}$, which computer should I use?

II. Roofline Model

How to compute Operational Intensity from a given Kernel?

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

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) FLOP

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?

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

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]
```

= N MUL + (N-1) ADD

A. Amount of flops W ? For every `i,j`: `C[i,j] = C[i,j] + A[i,k]*B[k,j]` 2N-1 flop

Loop over N*N —> Total = $(2N-1)N^2$ FLOPs $\approx 2N^3$ 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	$\frac{2N^3}{8(2N^3+2N^2)} \approx \frac{1}{8}$

II. Roofline Model

Question 3: Roofline Model (30 points)

Given the following code:

```
1  float A[N], B[N], C[N];
2  ...
3  const int P = 2;
4  for (int i = 0; i < N; ++i) {
5      int j = 0;
6      while (j < P) {
7          A[i] = B[i] * A[i] + 0.5;
8          ++j;
9      }
10     C[i] = 0.9 * A[i] + C[i];
11 }
```

a) What is the floating point operational intensity of the code? State all the assumptions you made and show your calculations.

b) For a peak performance of 409.7 GFLOP/s (single precision) and a memory bandwidth of 34 GB/s, find all positive P for which the code is memory bound. Assume an infinite cache and state further assumptions you made. Show your calculations.

c) Draw below the roofline corresponding to (b) and label the axes.

OI with assumptions as we showed before

What is memory bound?