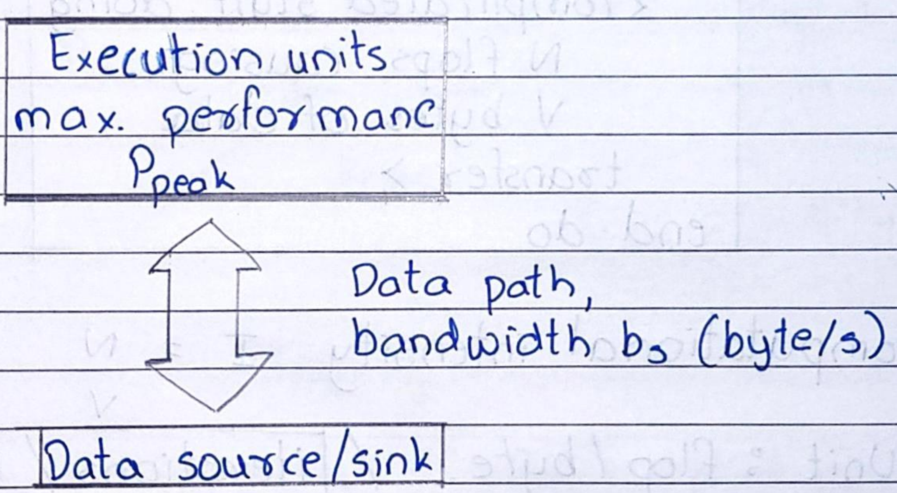
Rajat Munavalli
TY IT 21610069
Computer Algorithm Lab

Naive Roofline Model. (Simple Roofline Model)

① The naive roofline model is probably the simplest but still useful performance model for steady - state loops in high performance computing.

② Hardware view.



Hardware is viewed as two units

i) Execution unit of processor (at max performance)
units : mega flops / s or mega loops / s or iteration / s

ii) Data Source / Sink :
Main memory interface which can store data or deliver data at a maximum speed (bandwidth) (b_s)
unit : byte / s
Not necessary it should be memory interface.

③ Software view :
Comprises of several back to back loops which is sufficiently large to startup and winddown effects like pipelining, prefetching. i.e steady state behavior

```
! may be multiple levels
do i = 1, <sufficient>
   <complicated stuff doing
       N flops causing
       V bytes of data
     transfer >
end do
```

Computational intensity $I = \frac{N}{V}$

Unit : flop / byte. / $\boxed{\text{iterations / byte}}$

④ There are possibly two situations that causes delay in a process :

a) The execution work limited to max performance of execution ~~usi~~ units.

b) The bandwidth of data path.

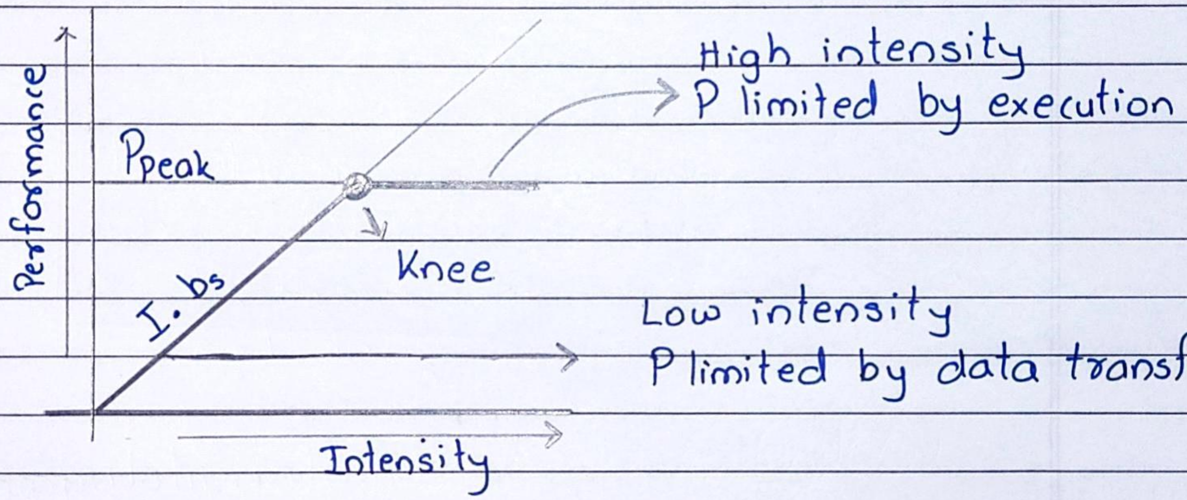
P_{peak} [flop/s]

$I \cdot b_s$ [flop / byte × byte / s]

∴ At any time the upper limit of the final performance is ~~Ppeak~~ minimum of Ppeak and I. bs.

$$\therefore \quad P = \min(P_{peak}, I \cdot b_s)$$

⑤ Graphical representation



The intersection of Ppeak and I. bs is called as "Knee" which is the point where best use of resources is observed. i.e max performance.

$$\therefore P_{max} = I \cdot b_s$$

The model relies on several assumptions, including perfect overlap of data transfers and computation, ignoring latency effects and assuming steady-state code execution. Overall the naive roofline model provides a simplified way to analyze the potential performance of a code on specific hardware platform, helping developers understand whether their code ~~an a specific~~ is limited by computationa~~l~~ or data transfer and guiding optimization efforts to achieve better performance.