



Технології графічного процесінгу

Лекція 1: Вступ

Кочура Юрій Петрович
iuriy.kochura@gmail.com
[@y_kochura](#)

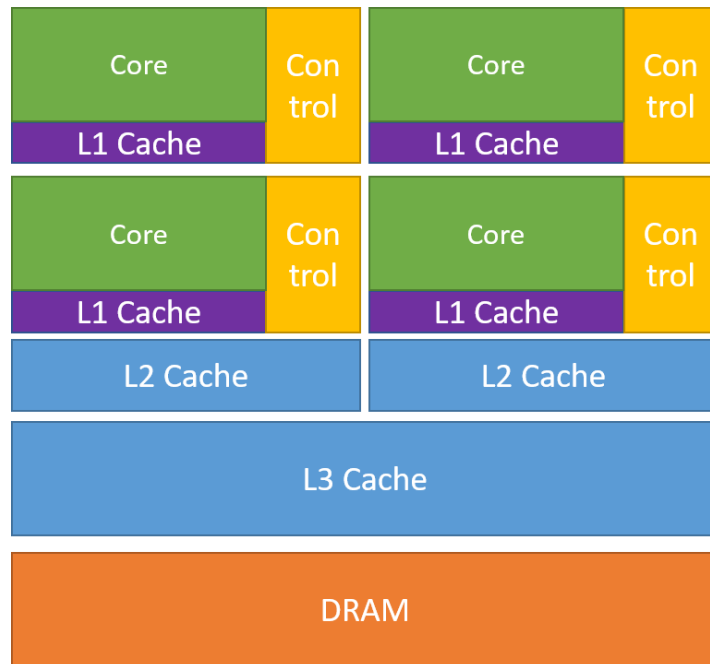
Сьогодні

- Будова CPU та GPU
- Переваги використання GPUs
- Паралельний vs Розподілений
- Основні закони паралельних обчислень
- Області застосування гетерогенних паралельних обчислень

Сьогоднішні виклики обчислень

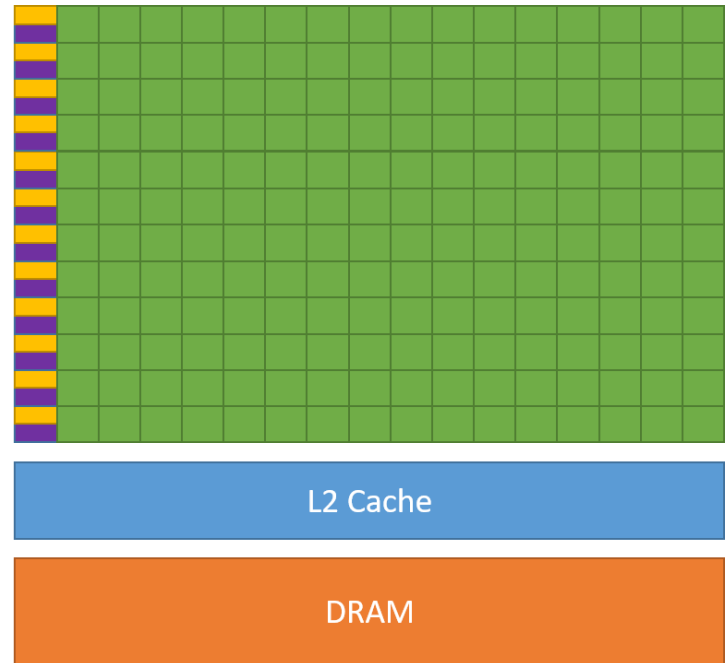
- Навички **виконання обчислень** є важливими для вивчення практично усіх дисциплін
- **Наука про дані** та **машинне навчання** стають основними навичками в більшості STEM
- Практично всі **процесори багатоядерні**, від мікроконтролерів до суперкомп'ютери
- Промисловість і наукові відкриття вимагають **ШІ** та **прискорених обчислень**

Переваги використання **GPUs**



CPU

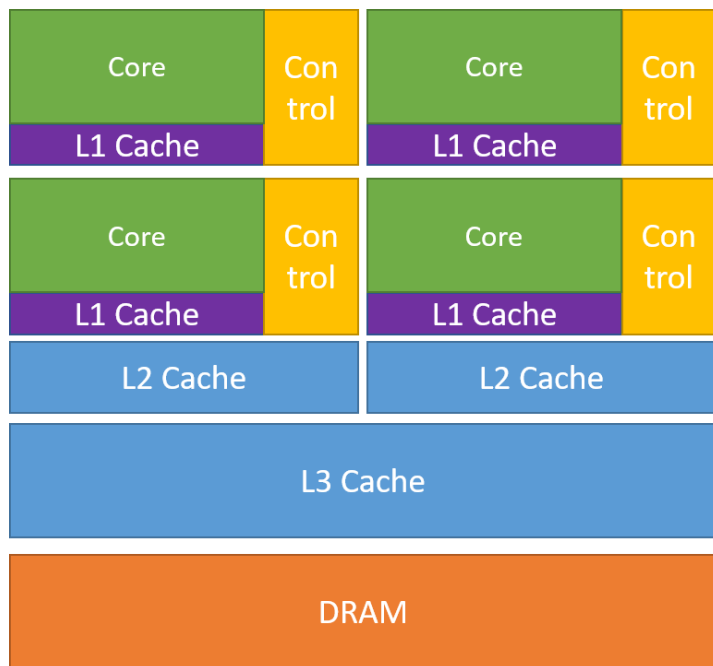
Central Processing Unit (CPU)



GPU

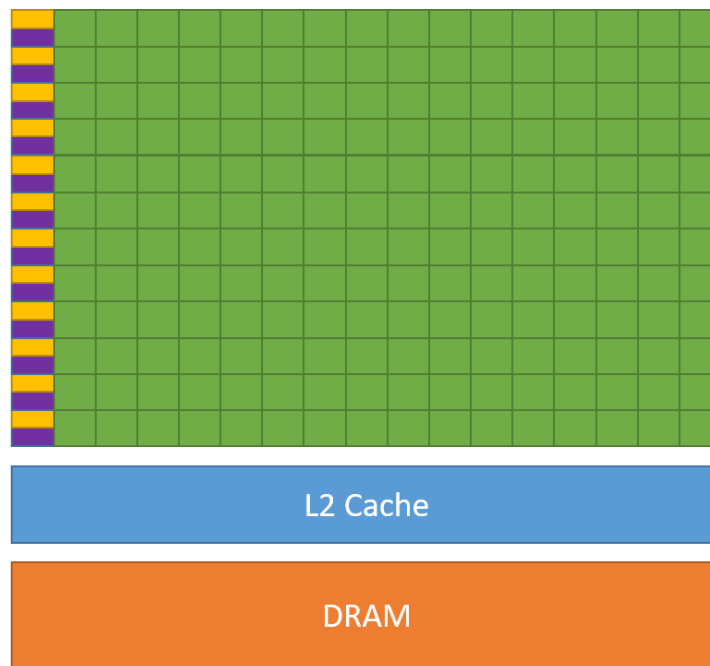
Graphics Processing Unit (GPU)

CPU та GPU спроектовані дуже по-різному



CPU

Ядра орієнтовані на зменшення затримок Latency
Oriented Cores



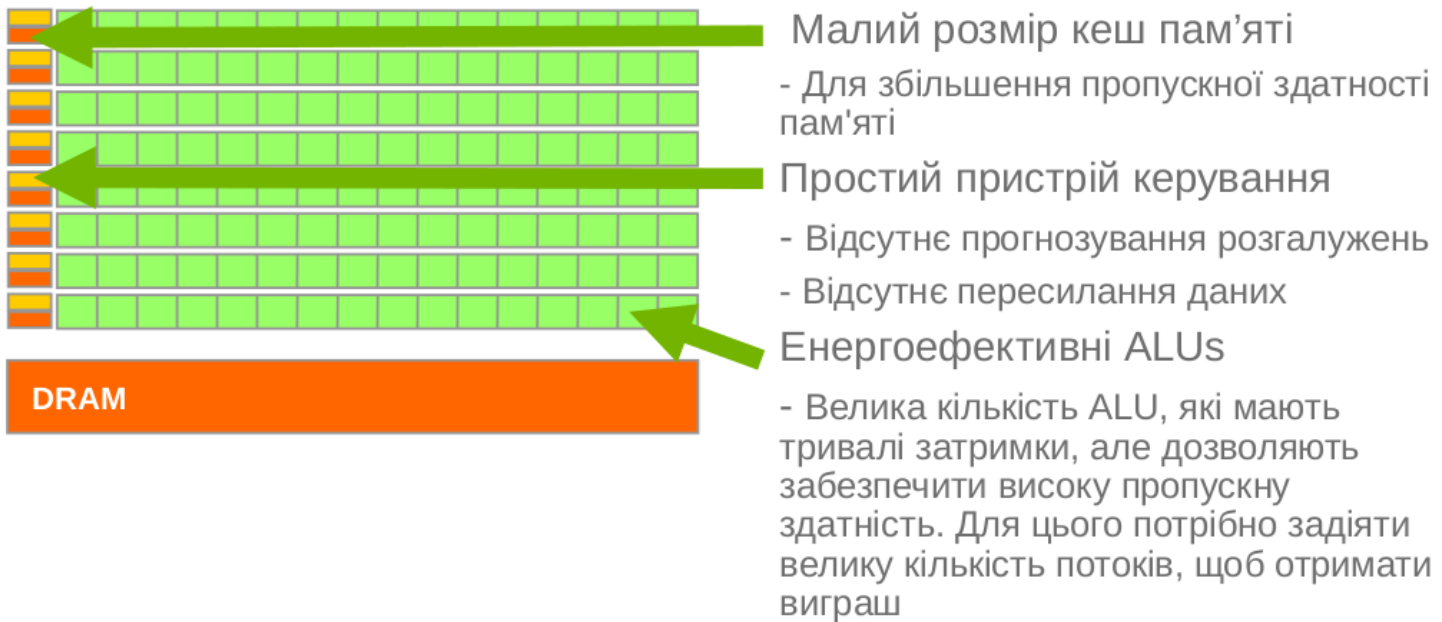
GPU

Ядра орієнтовані на збільшення пропускної здатності
Throughput Oriented Cores

CPU: Архітектура орієнтована на зменшення затримок



GPU: Архітектура орієнтована на збільшення пропускної здатності

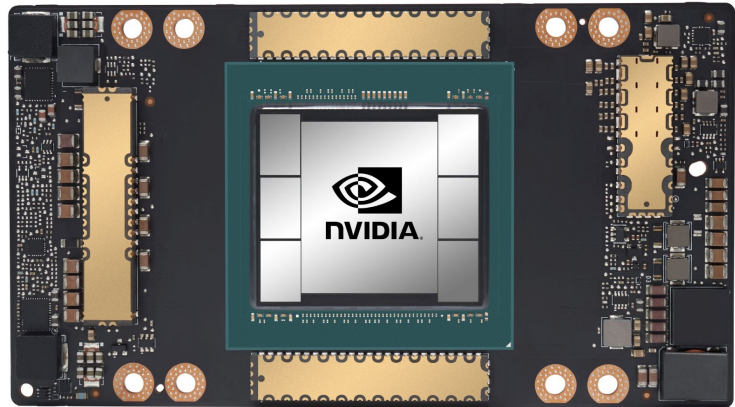


CPU vs GPU



Джерело: [12th Gen Intel Core](#)

- 12th Gen Intel Core
 - 14 Processor Cores
 - 10 nm process



Джерело: [NVIDIA Ampere Architecture In-Depth](#)

- NVIDIA A100
 - 6912 CUDA Cores
 - 7 nm process

Програми-переможці використовують як **CPU** так і **GPU**

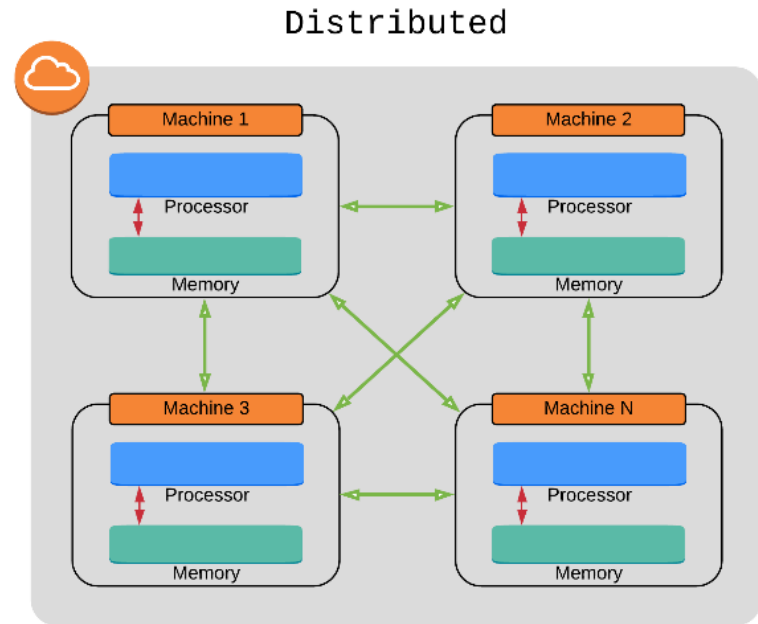
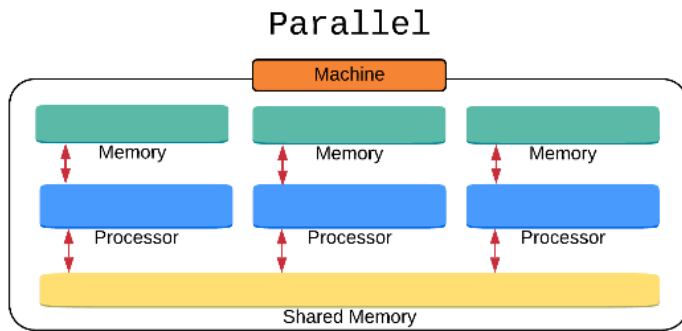
CPU використовують для
послідовних частин коду, де
затримки мають значення

CPU може бути в $\times 10+$ швидшим ніж GPU для
частин програми, які виконуються послідовно

GPU використовують для
паралельних частин коду, де
пропускна здатність виграє

GPU може бути в $\times 10+$ швидшим ніж CPU для
частин програми, які можуть бути виконані
паралельно

Паралельний vs Розподілений

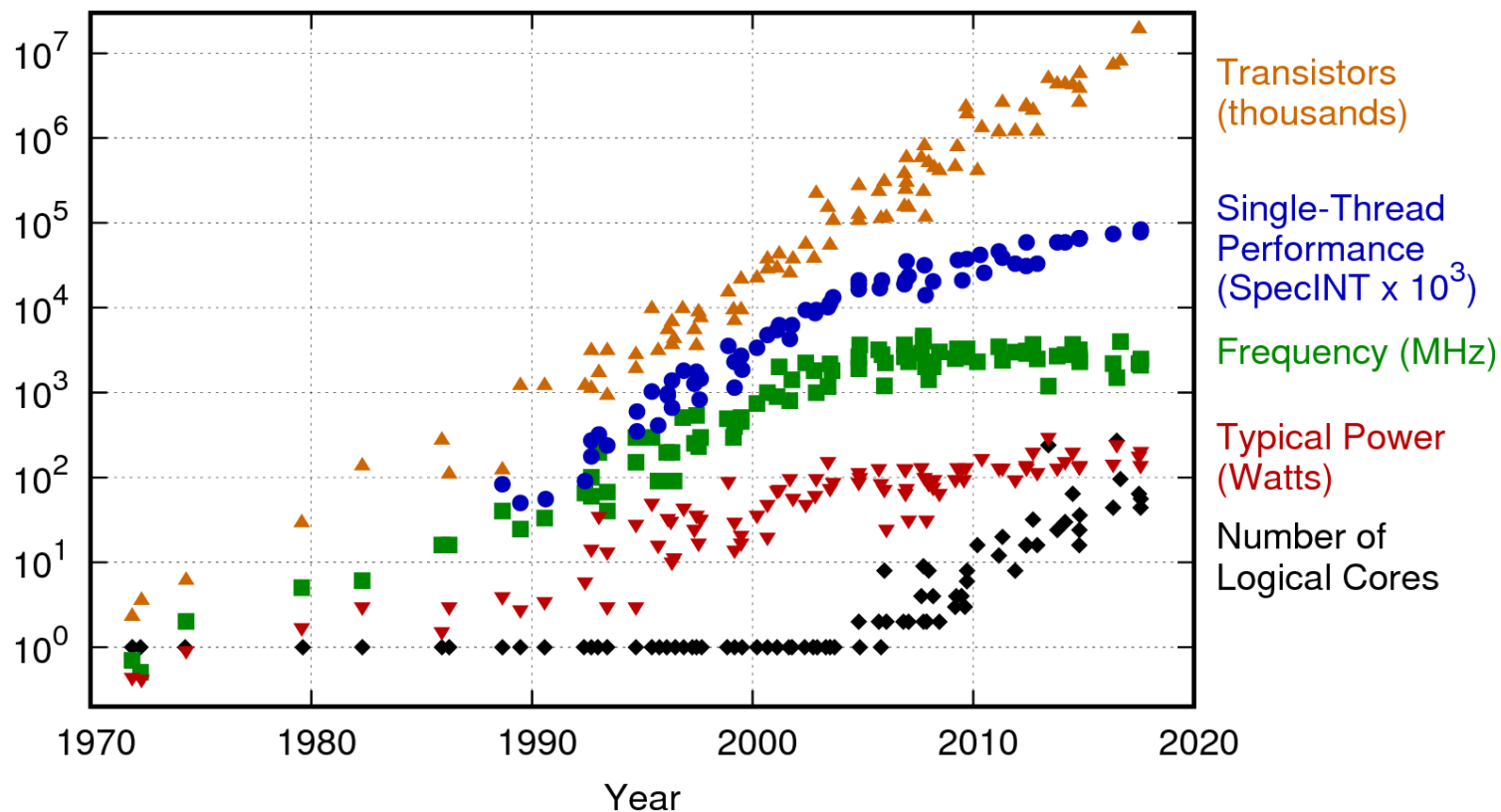


Паралельні обчислення — обчислення, які виконуються одночасно на кількох процесорах в одній системі — багатопотоковій.

Розподілені обчислення — використання кількох процесорів на кількох машинах, які спілкуються через мережу.

Чому паралельні
обчислення важливі?

42 Years of Microprocessor Trend Data



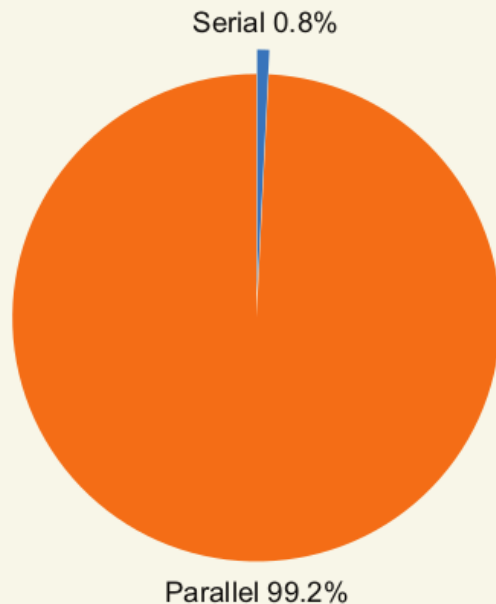
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Example

Let's take a 16-core CPU with hyperthreading and a 256 bit-wide vector unit, commonly found in home desktops. A serial program using a single core and no vectorization only uses 0.8% of the theoretical processing capability of this processor! The calculation is

$$16 \text{ cores} \times 2 \text{ hyperthreads} \times (256 \text{ bit-wide vector unit}) / (64\text{-bit double}) = 128\text{-way parallelism}$$

where 1 serial path/128 parallel paths = .008 or 0.8%. The following figure shows that this is a small fraction of the total CPU processing power.



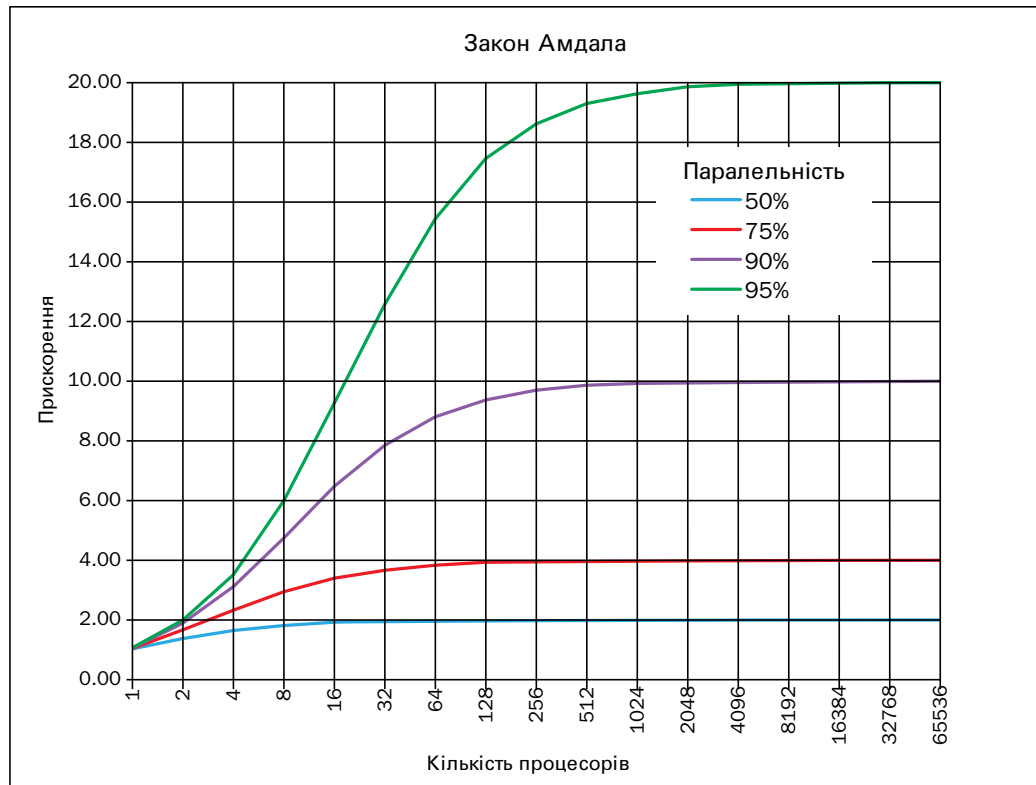
A serial application only accesses 0.8% of the processing power of a 16-core CPU.

Які потенційні переваги
паралельних обчислень?

- Прискорення часу виконання програм
- Підвищення енергоефективності
- Вирішення великих проблем

Основні закони паралельних обчислень

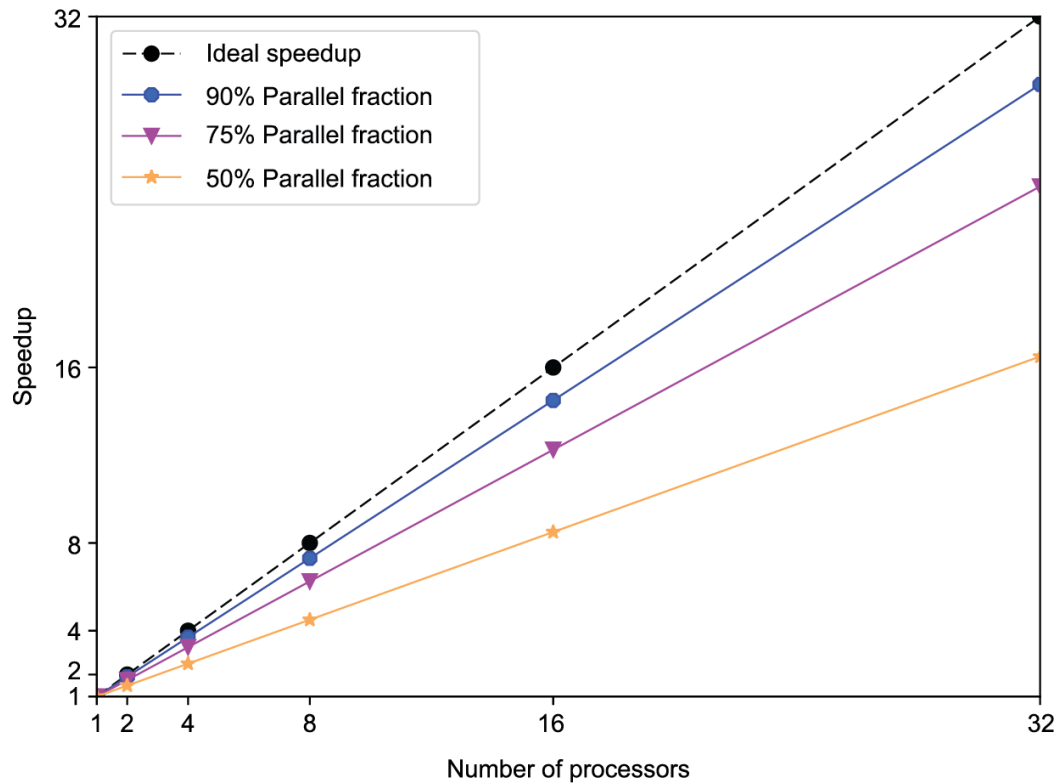
Межа для паралельних обчислень: закон Амдала



$$\text{SpeedUp}(N) = \frac{1}{S + \frac{P}{N}},$$

де N — кількість процесорів, S — частина яку можна виконувати послідовно,
 P — частина яку можна виконувати паралельно, $S + P = 1$

Прорив паралельної межі: закон Густафсона-Барсіса



$$\text{SpeedUp}(N) = N - S \cdot (N - 1),$$

де N — кількість процесорів, S — частина яку можна виконувати послідовно

Застосування

Застосування гетерогенних паралельних обчислень

**Машинне
навчання**

**Наукове
моделювання**

**Біомедична
інформатика**

**Обробка
відео**

**Фінансова
аналітика**

**Інженерне
моделювання**

**Медична
візуалізація**

**Астрономія
та астрофізика**



Mythbusters Demo GPU versus CPU



Share



Mythbusters Demo GPU versus CPU

Кінець 🏁