Sequential vs Parallel Processing

26 October 2025 11:54

Pre-req

Sequential Processing

- Tasks which are executed one by one or say one after the other
- Basically each task starts only after the previous one has completed.

Ex: Common in single core - CPUs

Parallel Processing

In parallel processing, multiple tasks are executed simultaneously by distributing the workload across multiple CPUs cores, GPUs or any other resources.

- Basically tasks are divided and assigned to a different processing units to run concurrently

Sequential Processing

Parallel Processing

→ How many tickets can be booked at once

3

Feature	Sequential Processing	Parallel Processing
Execution Style	Step-by-step (one at a time)	Simultaneous execution
Resource Usage	Single core/thread	Multiple cores/threads
Speed	Slower for large tasks	Faster for large/divisible tasks ← GPUs ⇒ Matrix Multiplication
Complexity	Easier to implement (Nokia 1600)	More complex (sync, data sharing, etc.) (iPhone)
Task Dependency	Suitable for dependent tasks	Best for independent tasks
Scalability	Limited	Highly scalable
Example	Simple loop iteration	Matrix multiplication, image processing

low budget is required

More compute power
↓
More budget for a project

DO IT AT YOUR OWN RISK

Press **windows + R**-->
write **cmd**-->
and copy paste the below command:
wmic cpu get NumberOfCores,NumberOfLogicalProcessors

```
C:\WINDOWS\system32\cmd.
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thesi>wmic cpu get NumberOfCores,NumberOfLogicalProcessors
NumberOfCores  NumberOfLogicalProcessors
14             18
```

Core 2 Duo vs Dual Core } two cores

is a branding term given by Intel.

'General terminology'

60 seconds → 'out of available processing capacity'

Utilization	Speed		Base speed:	3.60 GHz
7%	1.43 GHz		Sockets:	1
			Cores:	14
Processes	Threads	Handles	Logical processors:	18 → Hardware Thread
373	7182	189075	Virtualization:	Enabled
			L1 cache:	1.4 MB
Up time			L2 cache:	14.0 MB
0:16:51:36			L3 cache:	18.0 MB

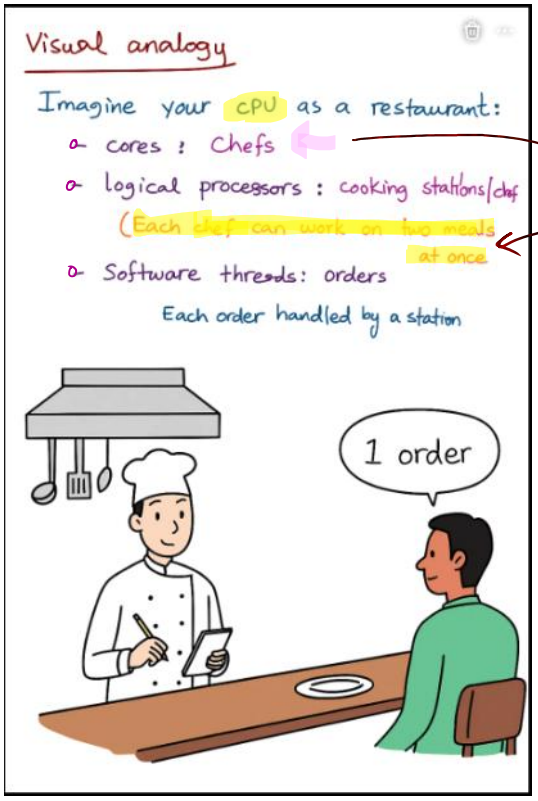
Software Threads

Core vs Thread

Core: is a physical processing unit within the CPU and each core can independently execute instructions and run tasks

Thread: is a virtual processing unit managed by CPU and each core can run one or more threads.

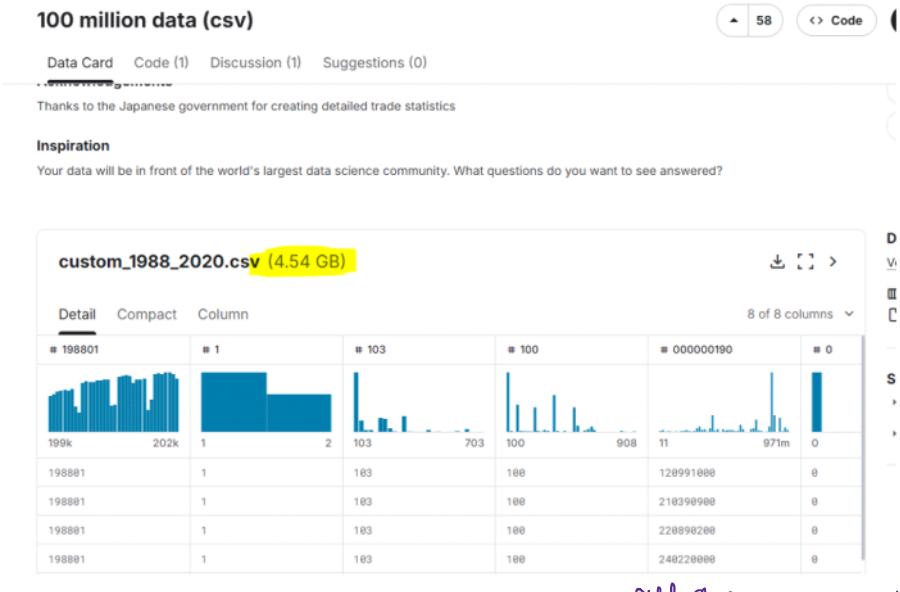
Note: Most modern CPUs support Simultaneous Multi-Threading or Hyper-Threading (HT)



TASK: Let us say if you are working with large datasets (~100M rows) on a 16GB RAM or lesser RAM and 256 GB SSD.

↓

Can the machine handle this??



→ out of memory error (OME)

https://www.kaggle.com/datasets/zanjibar/100-million-data-csv

- load the data & do typical EDA

+ Pro-tip

Best practices to handle big data