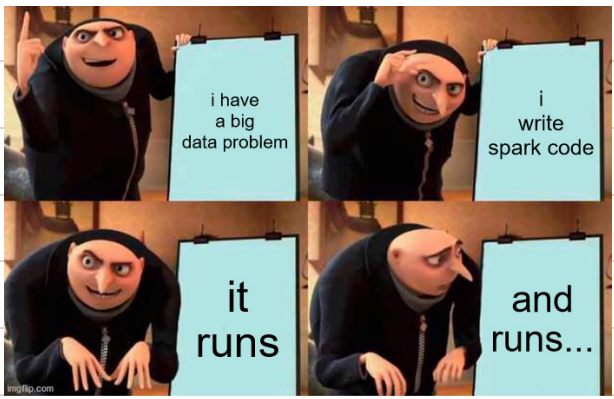
Session 10

Apache Spark

June 30, 2025



Issue With Pandas

TBs

1. It tries to load everything in RAM
2. Single Core
3. Does not use distributed Computin

```
id,name,age,salary
1,Alice,29,100000
2,Bob,31,95000
....
```

UTF-8

every character -> 1 byte

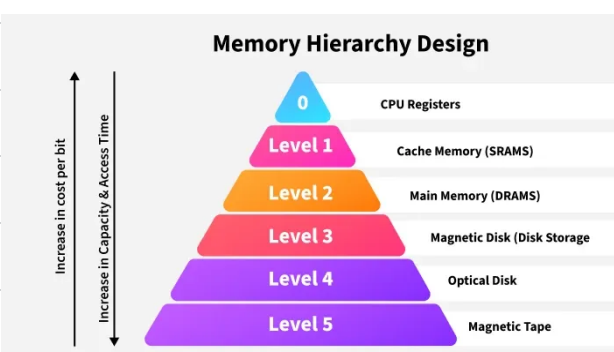
+

operational + structural overhead

float 64
float 32
-
.
.
.
1
s

10n space

Swap Space



Using `chunksize` parameter in `read_csv()`

For instance, suppose you have a large CSV file that is too large to fit into memory. The file contains **1,000,000 (10 Lakh) rows so instead we can load it in chunks of 10,000 (10 Thousand) rows- 100 times rows i.e** You will process the file in **100 chunks**, where each chunk contains 10,000 rows **using Pandas like this:**

```
import pandas as pd

# Load a large CSV file in chunks of 10,000 rows
for chunk in pd.read_csv('large_file.csv', chunksize=10000):
    print(chunk.shape) # process the shape of each chunk
```

Output:

Load a Massive File as small chunks in Pandas

https://www.geeksforgeeks.org/how-to-load-a-massive-file-as-small-chunks-in-pandas/

For optimize → chunk size

→ Quantization

Float 64 → Float 16

Model Variant	Parameters	VRAM (FP16)	VRAM (4-bit Quantization)
DeepSeek-LLM 7B	7 billion	16 GB	4 GB
DeepSeek-LLM 67B	67 billion	154 GB	38 GB
DeepSeek V2 16B	16 billion	37 GB	9 GB
DeepSeek V2 236B	236 billion	543 GB	136 GB
DeepSeek V2.5 236B	236 billion	543 GB	136 GB
DeepSeek V3 671B	671 billion	1,543 GB	386 GB

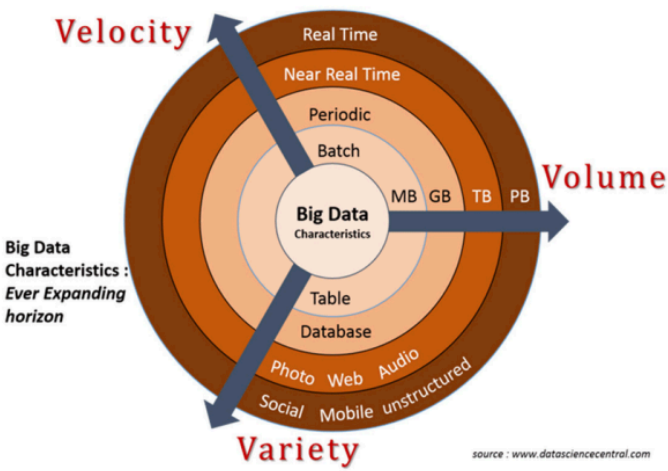
https://api.proxpc.com/media/uploads/2025/01/28/vram-deepseek-table-1.png

DASK

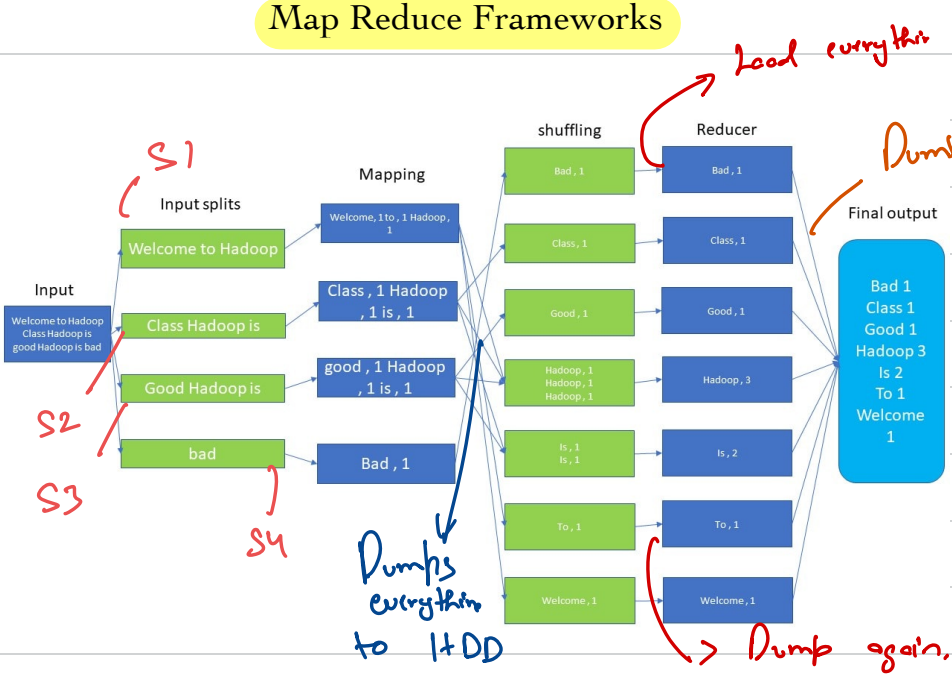
→ 70% of API / function is similar to Pandas

1. Does chunk wise processing
   100 GB - File
   → Load 16 GB
   → Dump result
   (repeat)
2. Lazy Evaluation
3. Distributed Computing. (not very stable)

Big Data


Velocity
Real Time
Near Real Time
Periodic
Batch
Volume
MB
GB
TB
PB
Big Data
Characteristics
Big Data
Characteristics :
Ever Expanding
horizon
Table
Database
Photo
Web
Audio
Social
Mobile
unstructured
Variety
source : www.datasciencecentral.com

Map Reduce Frameworks



Inverted Indexing → Google Stored.

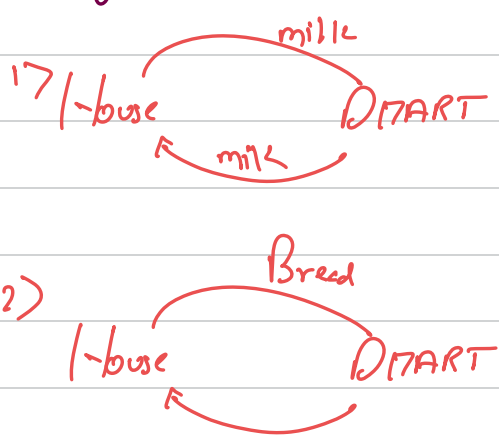
Apache Spark

1. Keeps Everything in RAM
2. Open Source
3. Distributed Computing
4. Lazy Evaluation

Groceries:

1. Milk
2. Bread
3. 2 kg potatoes
4. 1 kg basmati Rice

Eager Execution



Lazy Evaluations

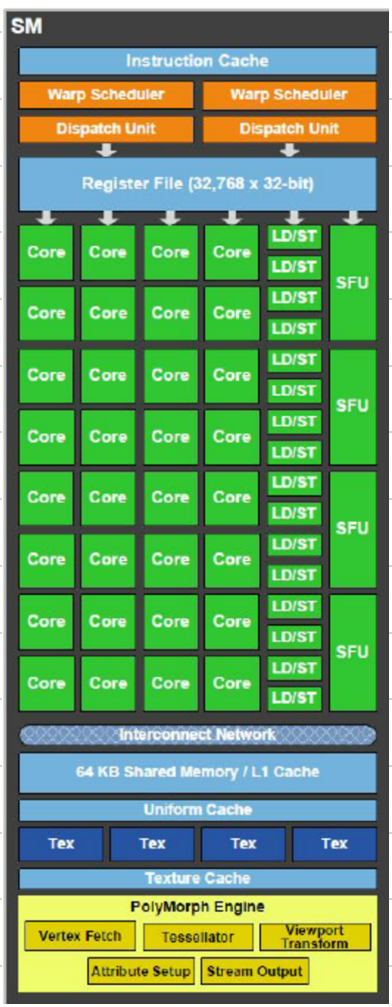
-> In End, I buy these stuffs

Java , C/C++

gcc some_file.c → compiled object

↓

Execution



The NVIDIA RTX 4090 (AD102, Ada Lovelace) features:

- **128 Streaming Multiprocessors (SMs)** techpowerup.com +12
- **16,384 CUDA cores** (derived from 128 SMs × 128 CUDA cores per SM) techpowerup.com +10

So in summary: **128 SMs**, packing a total of **16,384 CUDA cores**.

The RTX 5090 (based on the GB202 die, part of NVIDIA's Blackwell architecture) features:

- **170 Streaming Multiprocessors (SMs)** — each GPC has TPCs, and in total the GPU includes 170 SM units images.nvidia.com +14 .
- **21,760 CUDA cores** — that's 170 SMs × 128 CUDA cores per SM kitguru.net +6 .

So inside an RTX 5090 you get **170 SMs** and **21,760 CUDA cores**.

① Multiplication

② Exponent

↳ Dump entire data in VRAM

10 GB → VRAM

256 KB

CUDA Cor

411<

What is a key limitation of using pandas for Big Data processing that Apache Spark addresses more effectively?

0 users have participated

A	Pandas has more complex functions for simple data analysis tasks.	0%
✓ B	Pandas operates primarily on in-memory data and may struggle with datasets that exceed available RAM.	0%
C	Pandas has a more extensive library for machine learning algorithms compared to Spark.	0%
D	Pandas can only process data in structured formats like CSV or Excel.	0%

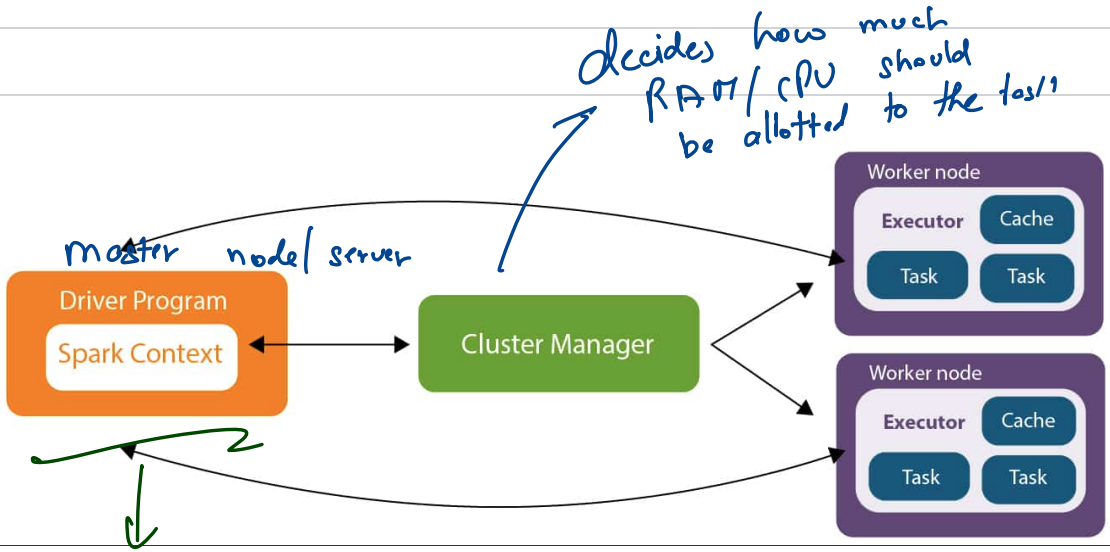
End Quiz Now

What is the primary advantage of using Apache Spark over traditional data processing tools for handling Big Data?

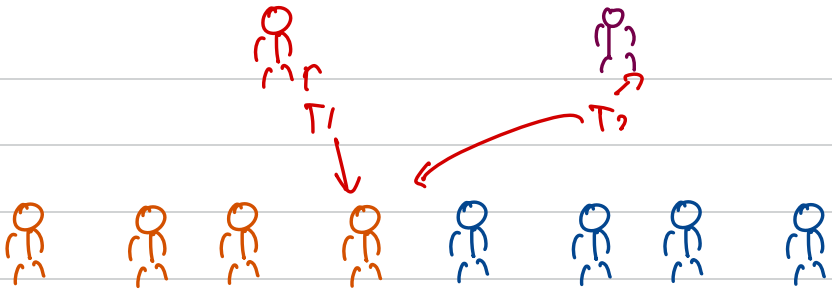
0 users have participated

A	Spark can only handle structured data types	0%
B	Spark requires a deep understanding of Java programming.	0%
C	Spark restricts the use of machine learning libraries.	0%
✓ D	Spark provides fast data processing for large, complex datasets.	0%

End Quiz Now



creates & assigns all data partitions to the slave/worker nodes



How does PySpark handle data partitioning by default when creating an RDD?

0 users have participated

A	PySpark keeps all data on a single partition to enhance data security.	0%
B	PySpark does not support partitioning; all data must be managed manually.	0%
C ✓	By automatically determining the number of partitions based on available resources and datasets.	0%
D	PySpark randomly scatters data across the network without regard for resource optimization.	0%

End Quiz Now