Big Data Systems

Guide to AI and Data Courses · Part IV: Data

The three Vs | Distributed storage | MapReduce and Spark | Stream processing | NoSQL and CAP | Neighboring courses

Sercan KÜLCÜ · Giresun University · Computer Engineering

© Sercan Külcü. All rights reserved. · Interactive HTML version of these slides: https://sercankulcu.github.io/files/ai_courses/slides/Chapter_27_Big_Data_Systems.html

INTRODUCTION

What Question Does It Ask?

> **How is data that does not fit on one machine stored and processed?**

Big data systems study how to store and process data that **does not fit on one machine**, or arrives faster than one machine can keep up, by spreading it over many machines. The course deals with the trade-offs between partitioning, replication, fault tolerance and consistency.

At a glance

APPROACH distributed storage and computation	**INPUT** huge, fast-flowing data	**OUTPUT** processed data, query results
SUCCESS MEASURE scalability, latency, cost	**MATH LOAD** low	**DATA NEEDS** very large

A Short History

2001
Laney describes the data management challenge as volume, velocity and variety (the 3 Vs).

2003
The Google File System paper: a fault-tolerant distributed file system built from cheap machines.

2004
Dean and Ghemawat publish MapReduce: a simple programming model across thousands of machines.

2006
Hadoop, an open-source counterpart of GFS and MapReduce, becomes its own Apache project.

2010
Spark is released as open source: keeping intermediate results in memory speeds up iterative jobs.

2011
LinkedIn open-sources Kafka: a distributed log system for event streams.

THE COURSE

Core Concepts

Volume, velocity, variety
The size of data, how fast it arrives, how many formats.

Distributed file system
Splitting files into blocks and replicating them across machines.

MapReduce
Map, shuffle, reduce: moving computation to the data.

In-memory processing
Spark: chaining steps without writing to disk.

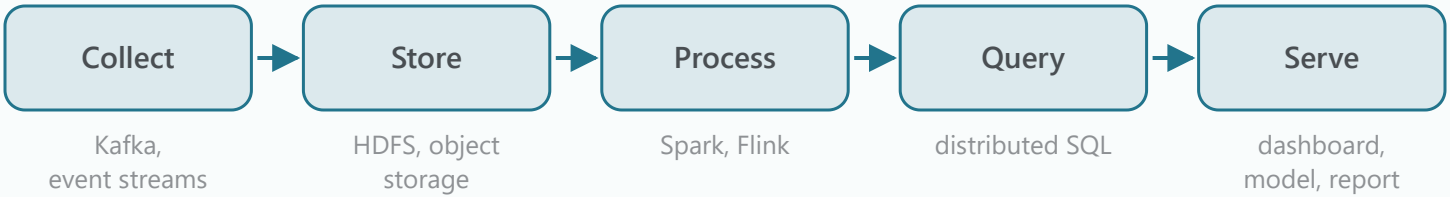
Stream processing
Processing an unbounded event stream as it arrives.

NoSQL
Key-value, document, wide-column and graph stores.

CAP theorem
Under a network partition, choosing consistency or availability.

Partitioning, replication
Spreading load and not losing data when a machine fails.

A Typical Workflow



Example: A video site collects billions of view events a day with Kafka, partitions them by date in object storage, computes nightly per-user viewing summaries with Spark and writes the results to tables that analysts can query with SQL.

THE COURSE

What Does the Course Cover?

1. The scaling problem: scaling up and scaling out
2. Distributed file systems: GFS and HDFS
3. The MapReduce programming model
4. Spark: RDDs, DataFrames and Spark SQL
5. Data models: relational, document, wide-column
6. Storage engines: LSM trees, B-trees, columnar files
7. Partitioning and load balancing
8. Replication and consistency models
9. CAP, distributed transactions, consensus: Paxos, Raft
10. Event streams: Kafka and publish-subscribe
11. Stream processing: windows, event time, Flink
12. Data lakes, warehouses and workflow orchestration

Skills you will gain

writing distributed jobs in PySpark | writing Kafka producers and consumers | columnar storage with Parquet | data modelling in Cassandra or MongoDB | diagnosing a slow Spark job

A typical semester; the order and coverage vary between universities.

TRY IT

Word Count over Search Logs

A site's search logs do not fit on one machine; worker machines must count together how often each word was searched. Go step by step: Map → Shuffle → Reduce. Change the number of workers and the combiner, and watch the time.

Map workers

W1
info host info
disk info node
cpu info query
host node info disk

W2
query info cpu
info disk disk
node host info
cpu query info info

W3
host disk node
info cpu host
query node info
disk info host cpu

Reduce workers

Next step ▶ | Restart

Workers = **3**

Combiner

Step 1/4: Split
Split: 29 ms · Map: 160 ms · Shuffle: 123 ms · Reduce: 116 ms
Elapsed time: **29 ms**

The text was split into 3 parts; each worker reads its own part.

A big data course teaches moving computation to the data: Map and Reduce tasks run independently, so adding workers speeds things up, but network traffic and the slowest worker set a limit. The **combiner** cuts the number of pairs on the network.

CONTEXT

Prerequisites

Courses worth knowing before this one, and why they are needed.

Course	Why it is needed
Operating Systems	processes, threads, file systems and the cost of I/O
Object Oriented Programming with Java	Hadoop, Spark and Kafka run on the JVM
Databases and SQL	the relational model, indexes, transactions

CONTEXT

Are You Ready for This Course?

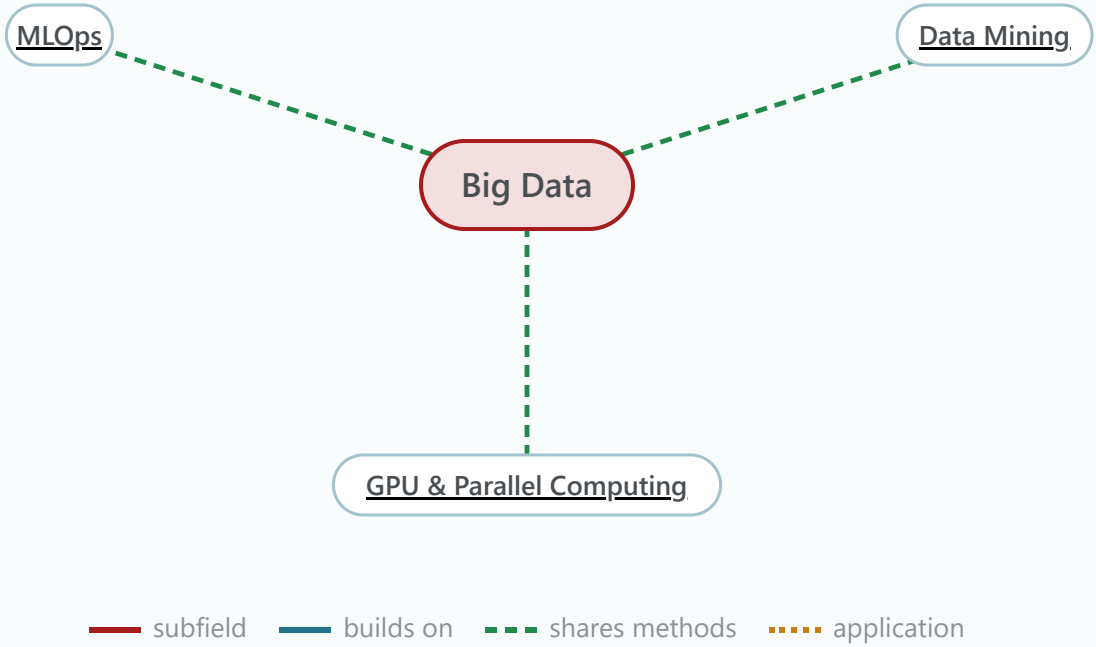
Ask yourself; for every "no", the right-hand column shows where to start.

Ask yourself	If not, start with
Can you explain the difference between a process and a thread?	An operating systems course: processes and concurrency
Can you write an SQL query with JOIN and GROUP BY?	SQL and the relational model from a database course
Can you write a program with classes and collections in Java or Python?	An object-oriented programming course in Java
Can you move files and run a script from the Linux command line?	A short introduction to the Linux command line

If you can say "yes" to most of these, you are ready; one or two gaps can be closed during the semester.

CONTEXT

Neighbouring Courses



OVERLAP

Overlapping Courses

Relation	Course	Shared ground
shares methods	Shares methods with Data Mining	scalable mining
shares methods	Shares methods with Machine Learning Systems and MLOps	data pipelines
shares methods	Shares methods with GPU Programming and Parallel Computing	parallel and distributed computing

DIFFERENCES

Where It Differs (1/2)

Criterion	Big Data	Data Mining	Data Science
Core question	How is data that does not fit on one machine stored and processed?	How are hidden, useful patterns found in large piles of data?	How do we ask the right questions of data and turn the answers into decisions?
Approach	distributed storage and computation	discovery-driven analysis	question, data, analysis, story
Input	huge, fast-flowing data	large tables or transaction data	raw, messy data
Output	processed data, query results	rules, clusters, outliers	report, visualization, model
Success measure	scalability, latency, cost	interestingness, support, confidence	impact on decisions, accuracy
Math load	low	medium	medium: statistics
Data needs	very large	large	moderate to large

- **Data Mining:** Data mining asks which patterns to look for; big data systems ask how that search runs on many machines.
- **Data Science:** Data science draws questions and conclusions from data; big data systems make that possible when data outgrows one machine.

Where It Differs (2/2)

Criterion	Big Data	GPU & Parallel Computing	MLOps
Core question	How is data that does not fit on one machine stored and processed?	How do thousands of cores share the same computation?	How is a model taken from the lab into production and kept running there?
Approach	distributed storage and computation	data-parallel computation	software engineering meets ML
Input	huge, fast-flowing data	parallelizable algorithm	trained model, data pipeline
Output	processed data, query results	accelerated program	a running, monitored service
Success measure	scalability, latency, cost	speedup, memory bandwidth	latency, uptime, data drift
Math load	low	low-medium	low
Data needs	very large	none	streaming production data

- **GPU & Parallel Computing:** A GPU speeds up computation with thousands of cores in one machine; big data systems spread data over machines, where network and disk are the bottleneck.
- **MLOps:** MLOps manages the model life cycle; big data systems build the data pipelines and storage that feed it.

PRACTICE

Applications

Log analysis
Errors and traffic from server logs.

Clickstream
Tracking user behavior in real time.

Recommendation backends
The histories of millions of users.

Internet of Things
Continuous measurements from sensors.

Financial transactions
Real-time fraud alerts.

Search index
Crawling and indexing web pages.

PRACTICE

Risks and Ethics

Silent data loss

Misconfigured replication or Kafka records deleted when retention expires can destroy data without anyone noticing.

Keep-everything habits

Collecting personal data because storage is cheap widens the breach surface; GDPR and similar laws require purpose limits.

Honouring the right to erasure

A person's data spreads into backups, replicas and derived tables; fully honouring a deletion request is hard.

Cloud bills and energy

A cluster left running or an unnecessary full table scan means a large bill and wasted energy.

In depth: Explainable AI and AI Ethics.

PRACTICE

Tools and Datasets

Tools and libraries

Apache Hadoop | Apache Spark | Apache Kafka | Apache Flink | Apache Cassandra | MongoDB | Apache Hive

Well-known datasets

Common Crawl | Wikipedia dumps | NYC TLC taxi records | GDELT | Google Books Ngram

PRACTICE

Common Misconceptions

Common belief	In fact
Big data just means a lot of data.	Velocity and variety matter as much as size; the real question is whether the data fits on one machine.
A distributed system is always faster.	On small data a single machine (pandas, SQL) is often faster; distribution has overhead.
Doubling the machines halves the time.	Shuffles, network traffic and the slowest task limit the speedup.
NoSQL has replaced SQL.	NoSQL offers different consistency trade-offs; SQL engines such as Spark SQL and Hive are widespread.

NEXT STEPS

After This Course

Courses that build on or extend this one

This course is mostly an end point: it builds on the others.

Related careers

Data engineer | Big data architect | Data platform engineer | Cloud engineer

Big data systems are the infrastructure of the data courses. From here you can move to MLOps, which takes models to production, to GPU and parallel computing, which speeds up computation, or to the data mining and data science that run on top of this infrastructure.

NEXT STEPS

Further Reading

- Kleppmann, *Designing Data-Intensive Applications*, O'Reilly, 2017.
- White, *Hadoop: The Definitive Guide*, 4th ed., O'Reilly, 2015.
- Chambers and Zaharia, *Spark: The Definitive Guide*, O'Reilly, 2018.
- Dean and Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", *OSDI*, 2004.

The End

Chapter 27: Big Data Systems

← Previous chapter · Map of the field · Next chapter →