CSCI E-103

Data Engineering for Analytics to Solve Business Challenges

Introduction to Data Engineering

Lecture 01

Anindita Mahapatra & Eric Gieseke

Harvard Extension, Fall 2025

Hello Everyone!

Anindita Mahapatra

- Graduate of ALM Management Program at HES
- BS & MS in Computer Science
- Solutions Architect at Databricks, Cloud-based Data & Analytics Company
- 25+ years of software and Big Data experience

amahapatra@g.harvard.edu
https://www.linkedin.com/in/aninditamahapatra/

Eric Gieseke

- Instructor for Software Design at the HES
- Graduate of ALM IT Program
- CEO & founder of Pago Capital, co-founder of Diyva
- 30+ years of software development experience

egieseke@g.harvard.edu
https://www.linkedin.com/in/ericgieseke/

Teaching Assistants

Ram Murali

Lead Solutions Architect, Databricks

rmurali@g.harvard.edu

https://www.linkedin.com/in/ramdas-murali-28a0081/

Paul Signorelli

Senior Solutions Architect, Databricks

psignorelli@gmail.com

https://www.linkedin.com/in/paul-signorelli-650a872/

Mohan Mathews

Lead Delivery Solutions Architect, Databricks

mohan.mathews@gmail.com

https://www.linkedin.com/in/mohan-mathews

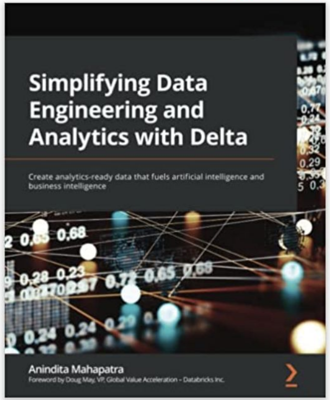
Agenda

- Course Logistics & Motivation for this course
- Big Data Ecosystem
- Data Personas
- Distributed Computing
- Machine Learning (ML) Ecosystem and its interaction with Data
- Industry trends in the Big Data space
- Business Justification for Tech Investment

- Lab
    - Introduction to Big Data Processing using Spark on the Databricks Platform hosted in the AWS cloud

Course Logistics

- Communication
  - Canvas
  - Slack
- Book
  - Simplifying Data Engineering and Analytics with Delta: Create analytics-ready data that fuels artificial intelligence and business intelligence
  - Spark - The Definitive Guide (Supplement)
  - Textbooks are available online and in the [Harvard Library]
- Labs
  - Databricks Platform ([Free Edition])
- Lecture - attendance is required
  - First Half: Lecture
  - Second Half: Lab
- Sections - attendance is recommended
  - Q/A session on Thrs (6-7 pm)
    - Help on assignments
    - Review course material
- Office Hours
  - Opportunity to ask/discuss 1-1
  - 30 min 1-1 slot - book through Calendly
  -
  - https://calendly.com/ramdaskm
  - https://calendly.com/psignorelli/office-hours
  - https://calendly.com/mohan-mathews/csci-e103-mohan-officehours
  - https://calendly.com/anindita-mahapatra/office-hours
  - https://calendly.com/eric_gieseke/office-hours



ISBN-13: 978-1801814867
ISBN-10: 1801814864

- Expectations
  - Remain engaged
  - Be inquisitive, ask questions
  - Explore and Learn, be open to ideas
  - Be Courteous & Professional
  - Lean in for Group Projects
  - Reach out to your team members early
    - Introduce yourself on slack

Date	Week	Lecture Topic	Assignment 1	Assignment 2	Assignment 3	Case Study 1	Assignment 4	Case Study 2	Final Project
Sep 2	1	Theory of Data Engineering	Introduction to Spark APIs						
Sep 9	2	Data Modeling & ETL							
Sep 16	3	Streaming Architectures		Data Ingestion and Exploration					
Sep 23	4	Data Lakes			Data Pipeline				
Sep 30	5	Change Data Capture							
Oct 7	6	Operationalizing Data Pipelines				TBD			
Oct 14	7	Data Warehouses					ML Pipeline		
Oct 21	8	Towards Reproducible Machine Learning							
Oct 28	9	MLOps Model Life Cycle Management							
Nov 4	10	Model Ensembles						TBD	
Nov 11	11	Data Imbalance							
Nov 18	12	Unstructured Data							Final Project
Nov 25 No Class Dec 2	13	Graph Analysis							
Dec 9	14	Continuous Improvement Cycle							
Dec 16	15	Class Presentations							Final Presentations

Assignments

4 Assignments (4 * 10% of grade)

2 Case Studies (2* 10% of grade)

Quiz-1 (4% of grade)

Quiz-2 (4% of grade)

Final Presentations (30% of grade)

Participation(2%) attendance, slack, recording viewing

Grading Policy

Submissions are due by midnight on the day they are due

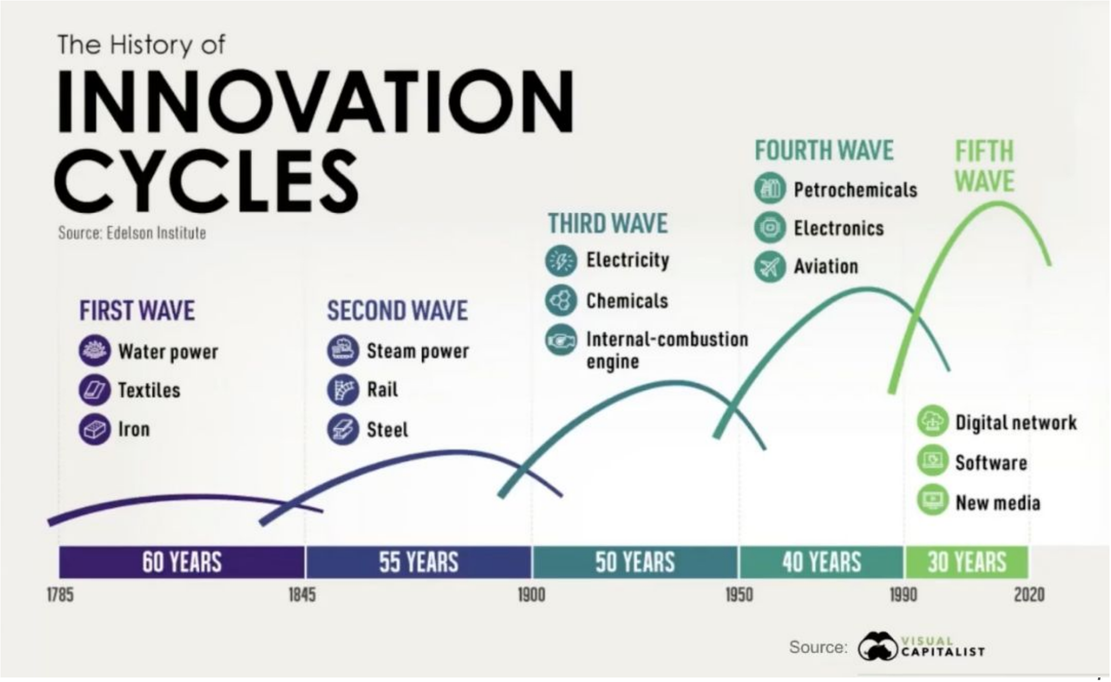
Late Policy: -2 points for every day late with a max of a 10 point deduction for lateness

[World Economic Forum](#) [#FutureOfJobs25](#) [https://www.weforum.org/videos/](#) [Report](#)

Which roles will see the fastest growth in demand by 2030?

Data Engineering in the Age of AI

[Video](#)



Motivation for Data Engineering

- We are at the interesting conjunction of **Big Data + Cloud + AI**
  - which is fueling so much of **innovation** all around us in every industry vertical
- Data is the **new oil** and is at the heart of every business
- Data drives ML which in turn gives businesses their competitive advantage
- Age of Digitization
  - Most successful businesses see themselves as tech companies first
  - Startups have the advantage of selecting the latest digital platforms
  - Traditional companies are all undergoing data digital transformations
- There is a lot of data around us, harnessing it makes it usable
- Technologies come and go, understanding the core problems is important
  - As technologists, we bring more impact when we align solutions with business challenges
- Speed to Insights is what all businesses demand and the key to it is data
- Data is as important an asset as code, so there should be governance around it
- Structured data is only 5-10% of enterprise data, the semi & unstructured data needs to be added to provide a holistic picture

Data drives business use cases in every industry

Threat
Detection
Prevention

Health and
Life
Sciences

Autonomous Vehicles

Connected Factory

Personalizations

Gaming/Entertainment

Smart Farming

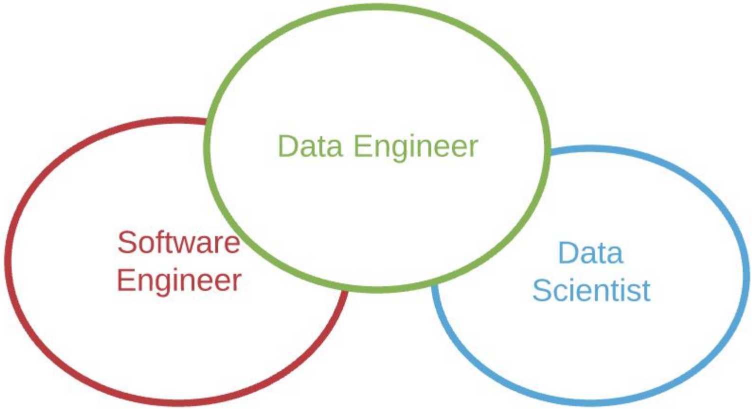
Banking

Forecasting

What is Data Engineering

Turning raw data to valuable insights

Data Engineering = Software Engineering + Lots of Data



Software Engineer	Data Engineer	Data Scientist
• Software design • Full stack development • Web/Mobile apps • DevOps • Service development	• Advanced data structures • Distributed computing • Concurrent programming • Knowledge of new & emerging tools: Hadoop, Spark, Kafka, Hive, etc. • Building ETL/data pipelines	• Data modeling • Machine learning • Algorithms • Business Intelligence dashboards

Data Personas with overlapping functions

- **Data Engineers** are focussed on maintaining the running of the data pipelines that ingest and transform data. This has a lot in common with a software engineering role coupled with lots of data.
- **BI Analysts** are focussed on sql based reporting and can be operational, financial, supply chain analysts
- **Data Scientists & ML Practitioners** are statisticians who explore and analyze the data (EDA, Exploratory Data Analysis) and use modeling techniques at various levels of sophistication
- **DevOps & MLOps** are focussed on the infrastructure aspects of monitoring and automation. MLOps is DevOps coupled with the additional task of managing the lifecycle of analytic models.
- **Data Leaders -** Chief Data Officers, Data Stewards are at the top of the food chain in terms of ultimate consumers of aggregated data

All these roles require an understanding of Data Engineering

Hardest Part of ML isn't ML, it's everything else

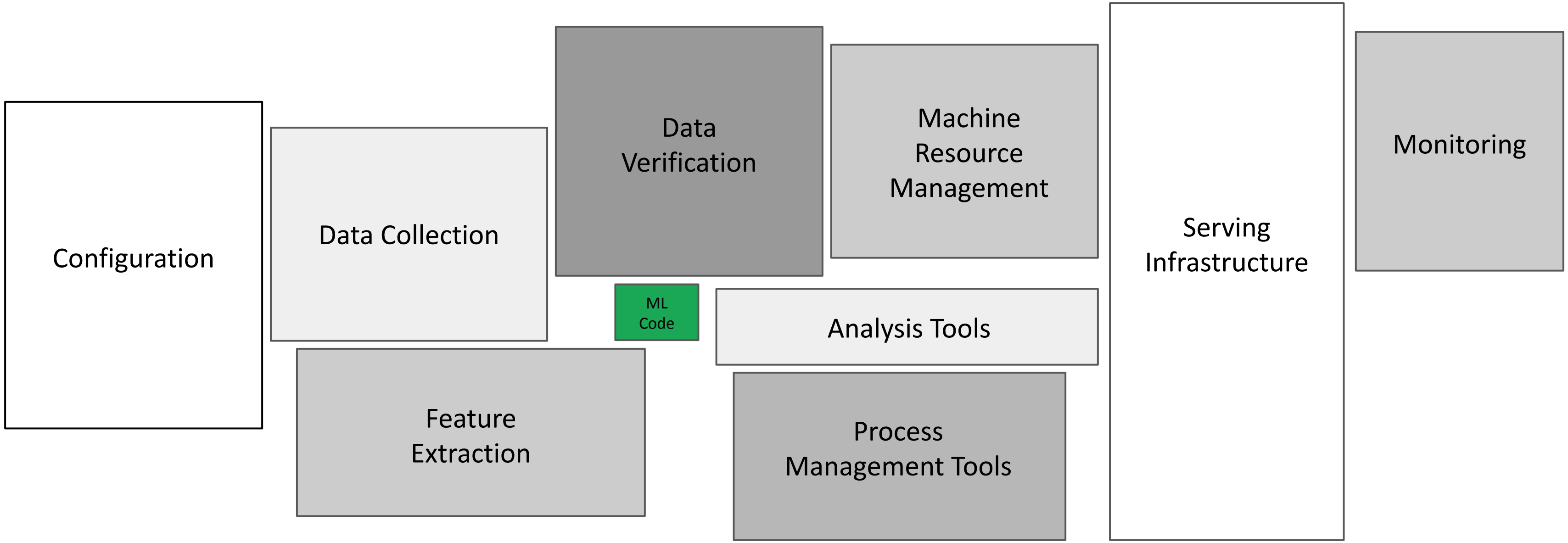


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small green box in the middle. The required surrounding infrastructure is vast and complex.

Big Data Characteristics

- Volume
  - Data Size, #Records, TPS
- Velocity
  - Batch, Realtime
- Variety
  - Structured, Semi-Structured, Un-Structured
- Veracity
  - Trustworthiness, lineage
- Value
  - Business Impact

Classifying data

<u>Size/Volume of the Data</u>

- Big Data - typically terabytes of data
  that cannot fit on a single computer node
- Single node data typically is considered modest data

<u>Variety aka Structure of data</u>

- Structured - Schema is well known and stable and hence assumed in the data
- Semi-Structured - Schema is built into the data Eg. XML, JSON format
- Un-Structured - Image, Audio, Video, documents, articles, tweets

<u>Velocity of Data</u>

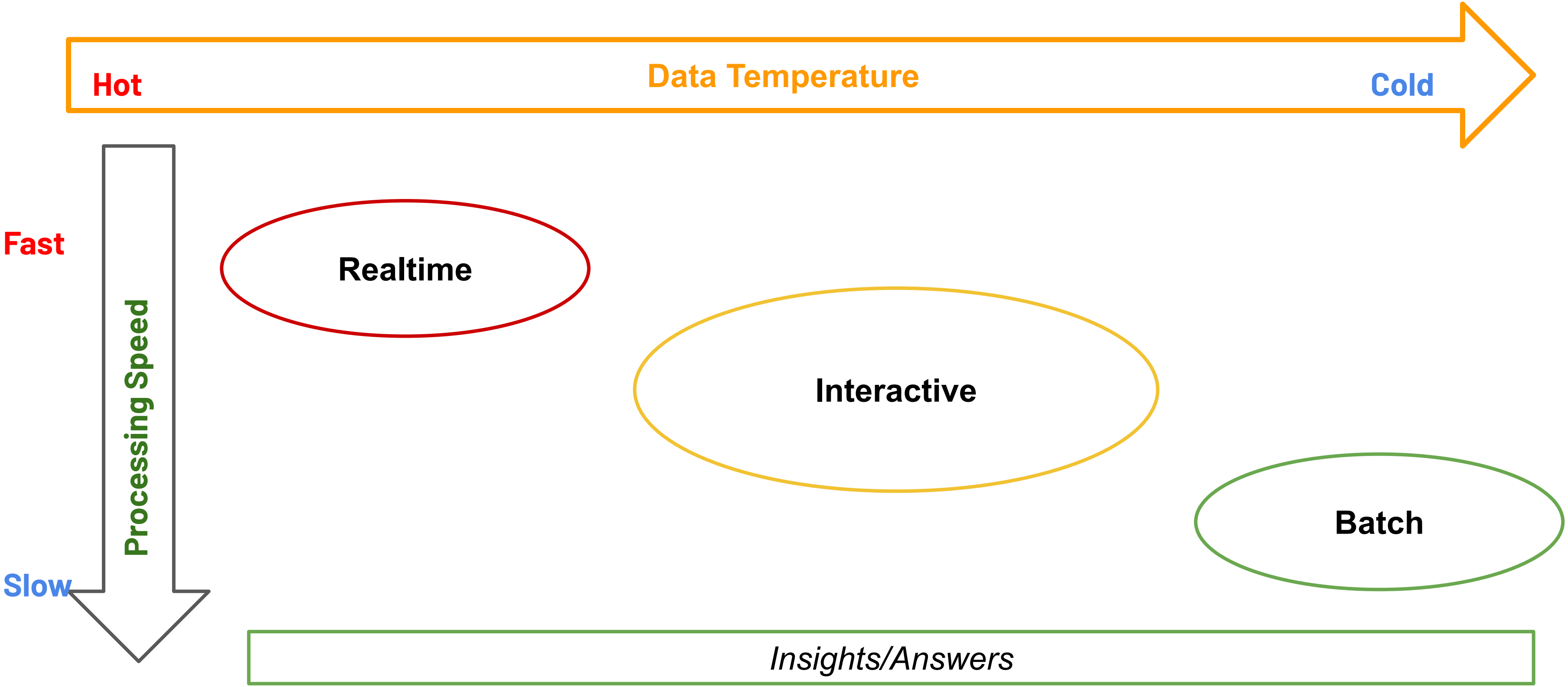
- Batch : Data arrives or is processed on a regular time frequency
- Streaming
  - Continuous: Data is processed as it comes
  - Micro-Batch: Data is aggregated in small micro batches typically a few second or millisecond.

<u>How often data changes</u>

- Hardly
  - Ex. demographic data
- Occasionally
  - Ex. Operational data
- Often
  - Behavioral data
- Frequently
  - Data associated with human sentiment/emotion,

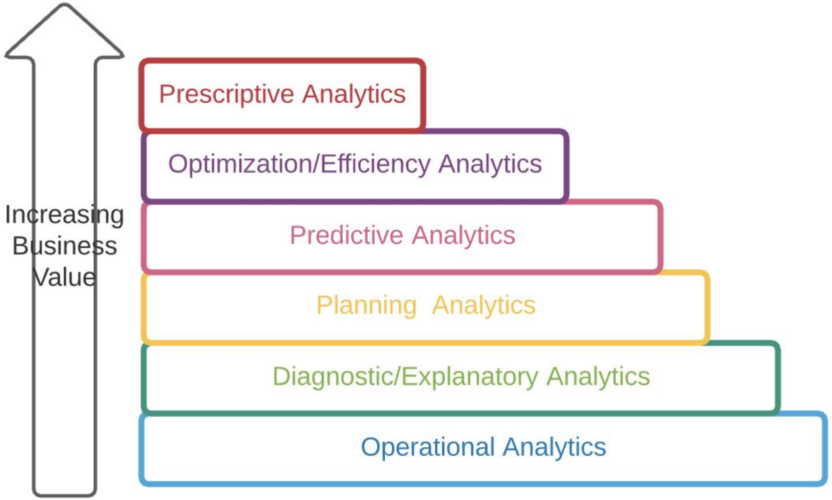
  sensor data

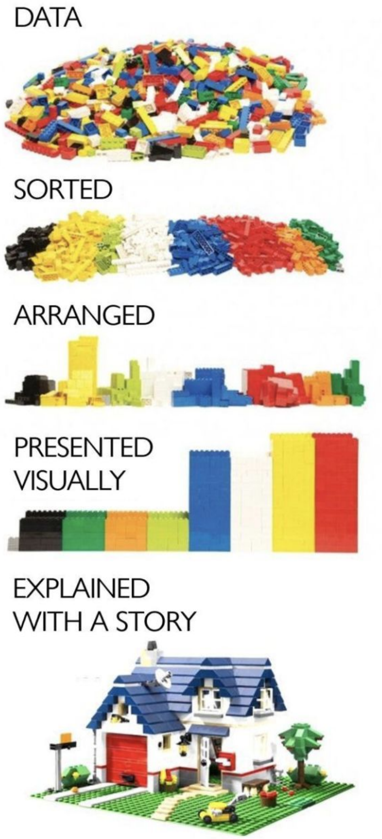
Data Storage/Compute Decisions



Data Analytics

From a LinkedIn post:
“The Difference between Raw Data and the Stories Data can tell.”





The analogy used is that of refining carbon to create a diamond.

Raw data is the carbon that gets increasingly refined.

The longer the processing layers, the more refined and curated is the value of the data.

However it is more time consuming and expensive to produce the artifact

Evolution of Data Platforms

1960s	1980s	2000s	2010s	2020s
Start of DBMS Technologies	**Data Warehouses**	**Web & Unstructured Data**	**Data Lakes**	**Lakehouse**
Staring with the flat files in the 60s and moving on to DBMS in 70s	The 1990s saw the rise of Data Warehouses, Dimensional Modeling, Data Marts This also saw the rise of MPP databases (such as Teradata) Expensive but **reliable** mainly for BI use cases with relational data on proprietary systems	Audio, Video Codecs exploded. Emphasis on Metadata grew. **Streaming** requirements surfaced NoSQL databases came to handle processing needs Hadoop came around the 2010s, open culture soared, business use cases suffered as data reliability dropped.	Spark increased in popularity and adoption because of speed and agility. Move to Cloud Data Platforms with **cheaper** storage. Specialized stores like graph DB continue to evolve. Focus on improving models - rapid strides in Deep Learning	Data Mesh, Data Fabric, Lakehouse are the newer entrants Focus on Data Domains & holistic Data Products Focus on data

SQL Vs NoSQL

SQL Based	NoSQL Based (Not just SQL)
ACID properties are honored in a transaction, namely ● Atomicity ● Consistency ● Isolation ● Durability	BASE properties are honored, namely ● Basically Available: The system is guaranteed to be available in event of failure. ● Soft State: State could change because of multi node inconsistencies ● Eventual consistency: All nodes will eventually reconcile o last state but there may be a period of inconsistency
Use cases with highly structured data with predictable inputs and outputs. Ex. financial system with money transfer where consistency is the main requirement.	Less structured scenarios involving changing schemas Ex. a twitter application scanning words to determine user sentiment. High availability despite failures is the main requirement

CAP Theorem

- Consistency: up-to-date and synchronized
- Availability:always get a response
- Partition Tolerance: system will operate even if some of its components are down

CAP theorem states that you can only have 2 of 3 of these.

Traditional Relational System support Consistency & Partition Tolerance at the expense of Availability
No-SQL Systems support Availability & Partition Tolerance at the expense of Consistency

It is harder to keep the data consistent as it grows

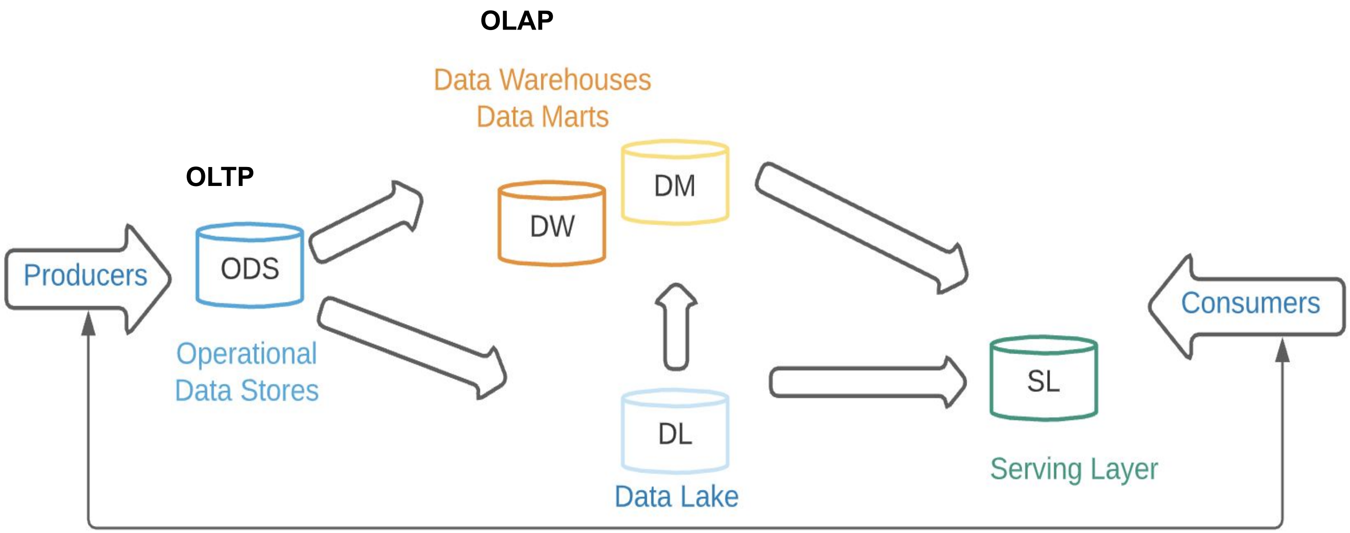
Operational (OLTP) Vs Analytic Data (OLAP)

OLTP: Online Transactional Processing
OLAP: Online Analytical Processing

	Operational Data (OLTP)	**Analytical Data (OLAP)**
NoSQL Technologies	Document Stores (Ex. MongoDB, Couchbase) Key value Stores (AWS S3, Azure Blob Storage) Column Family Stores (Ex.HBase, Cassandra)	Hadoop Systems Modern Cloud Data Platforms (Databricks, Snowflake)
SQL Technologies	Relational Databases (Ex. Oracle, SQL Server, MySql)	Relational Analytics Databricks, Snowflake

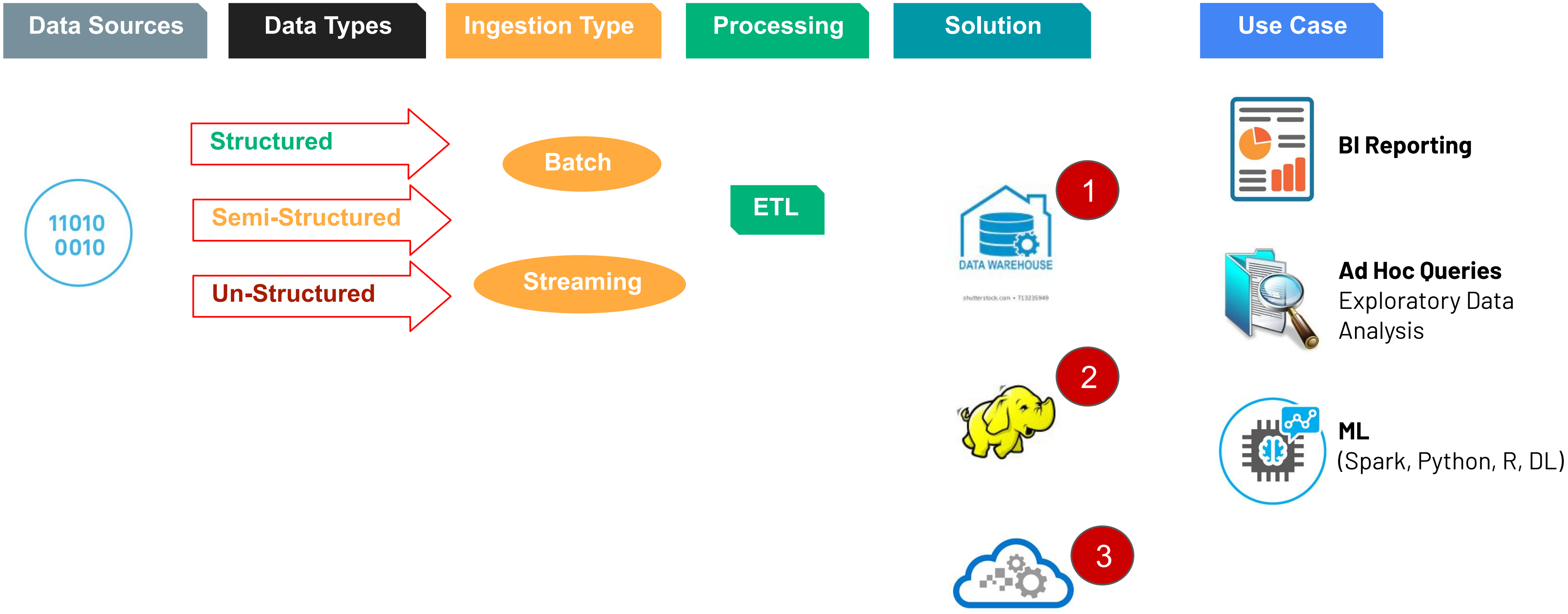
Data Producers & Consumers

ETL: Extract Transform Load (OLTP -> OLAP)
Reverse ETL: Online Analytical Processing (OLAP -> OLTP)



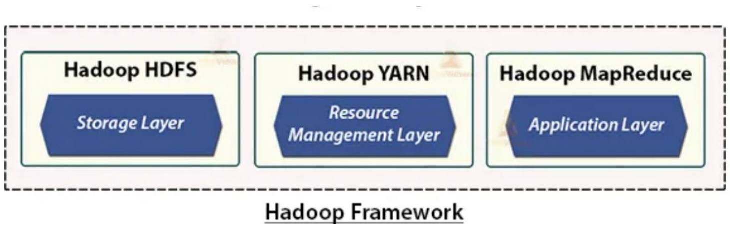
Different Consumers tap into the data at different stages

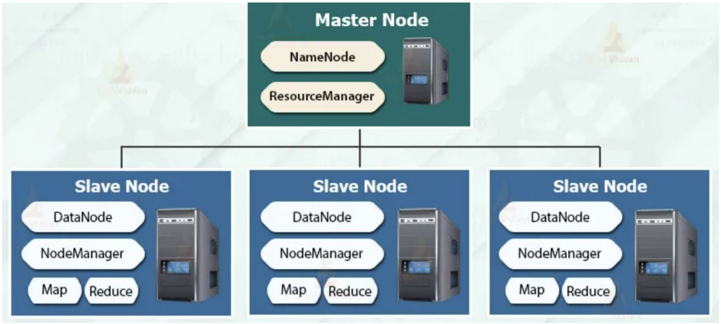
Evolution of the Modern Data Platform

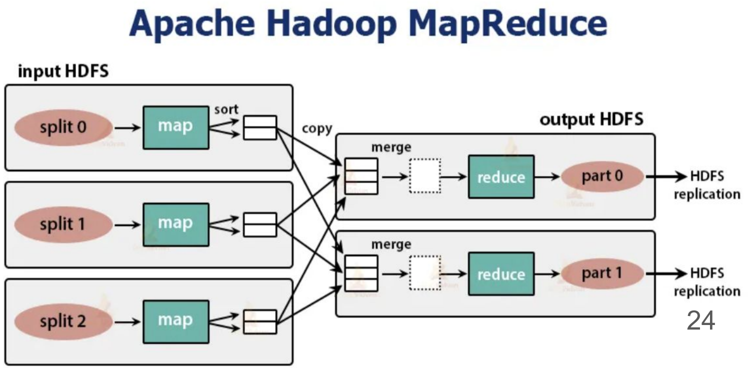


Hadoop

- Apache Open source project
  - Started as a Yahoo project in 2006
    - **Promise:** inexpensive, reliable, and scalable framework
  - Several distributions such as Cloudera, Hortonworks, MapR, EMR
  - Compatible with many types of hardware where it runs as appliances
  - Works with
    - Scalable distributed file systems like S3, and HDFS with triple replication(cheap storage)
    - Commodity grade hardware
    - Service oriented architectures with open source components
- Architecture
  - HDFS Data is broken into blocks, replicated a certain number of times and sent to worker nodes
  - NameNode keeps track of everything in the cluster
  - MapReduce sits on top of HDFS
  - JobTracker & TaskTracker monitor progress
  - YARN allocates resources that the JobTracker spins up and monitors
  - All the results from the MapReduce stage are then aggregated and written back to disk in HDFS



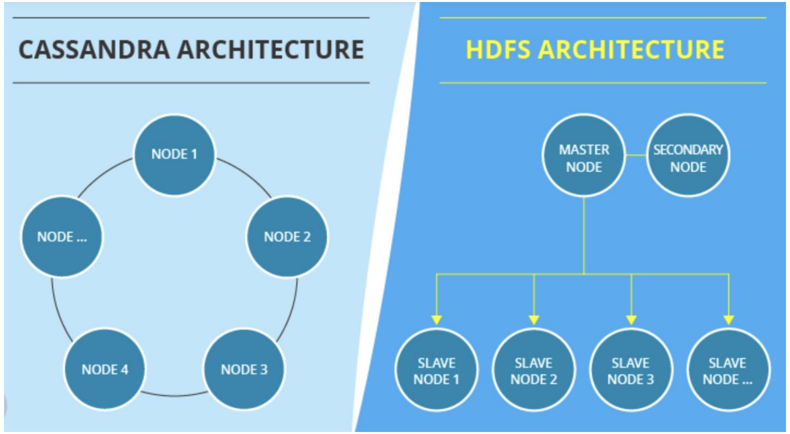




Relational Vs Hadoop Data Stores

	Relational	**Hadoop**
Data architecture and volume	Structured database approach in which data is stored in rows and columns which can be updated with SQL and presented in different tables.	Hadoop is not a database, but rather a distributed file system that can store and process a massive amount of data across computers. open source projects like Hive and Presto can abstract the file system into a table like format that is accessible with SQL.
Data Variety	Manage and process structured and semi-structured data in a limited volume	Manage and process all data types; structured, unstructured, and semi-structured data.
Technical Expertise	Most relational databases are *arguably* easier to use, fewer moving pieces in comparison	Managing cluster, the Hadoop nodes, security,
Security Issues	Well understood	Authentication and encryption modules through kerberos are harder to implement
Functional Issues	Supports tx and is used for BI reporting scenarios	Concept of write once read many hence not conducive for frequent updates

Hadoop's HDFS (OLAP) Vs Cassandra (OLTP)



Both deal with large data

HDFS has a master slave architecture and favors larger files

Cassandra is masterless, hence more resilient to failures & allows for varying levels of consistencies.

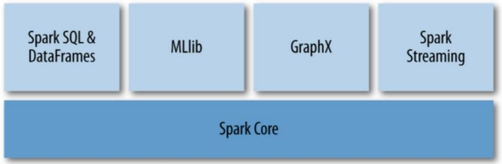
Hadoop supports partitions, Cassandra provides record level indexing

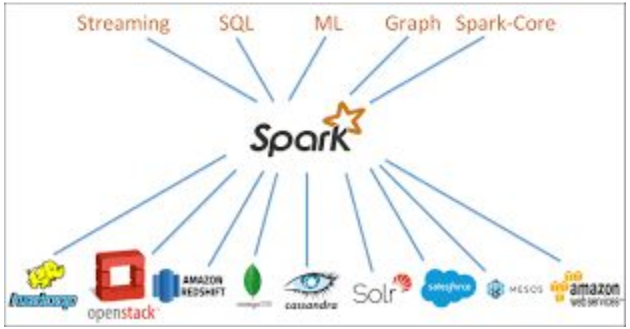
They can coexist where Hadoop is used for Data Lake and the Serving can be off Cassandra

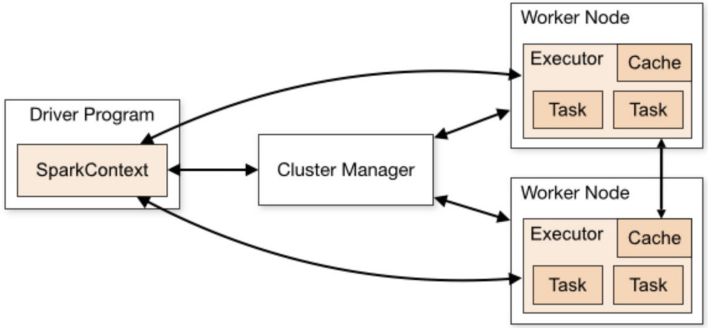
Both HDFS & Cassandra adhere to CAP, supporting Availability and Partition Tolerance.

Spark: A unified analytics engine for large-scale data

- Apache Open source project
  - Started in 2012, at the AMPLab at UC Berkeley.
  - Written in Scala and has support for
    - Scala, Java, Python, R, SQL
  - Connectors for several disparate providers/consumers







Hadoop Vs Spark

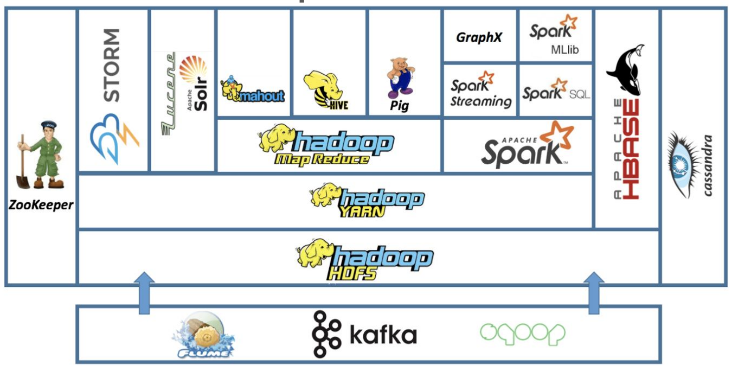
Spark is ~100x faster

- Hadoop: Chains of map and reduce - each of which goes to disk
- Spark processes and retains data in memory
    - for subsequent steps in a DAG (Directed Acyclic Graph)
    - Directed acyclic graph of all the operations/transformations

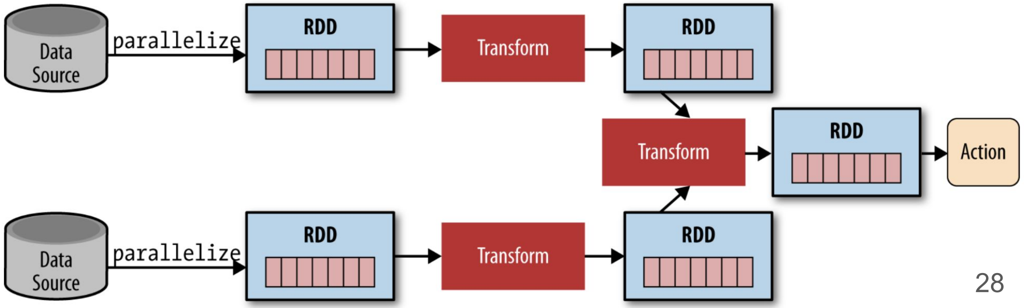
Data

- Hadoop reads and writes files to HDFS,
- Spark processes data in RAM with occasional spill to disk
    - RDD: Resilient Distributed Dataset: **immutable distributed collection of elements of data**
    - DataFrames & DataSets are newer abstraacions to RDD

Hadoop Stack



Spark Processing



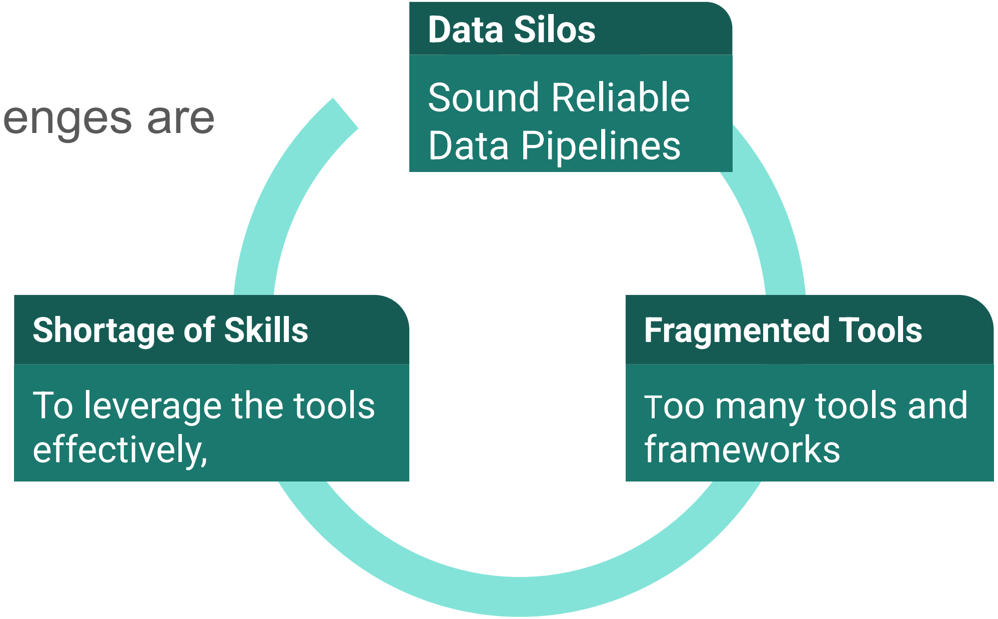
Data Platform Models

SaaS	Hosted Applications
PaaS	OS, Development Tools, Database Management, BI analytics
IaaS	Data Center, Networking, Servers & Storage

	What is it	Benefits	When to use
SaaS Software as a Service	Delivers applications from third-party vendors for use on-demand over the Internet	-Ease of Use -Payment Flexibility -Easy Customization	if you want an app but do not have the time or resources to build or manage the software.
PaaS Platform as a Service	Offers a platform on which a developer can design and deploy an application without getting involved in time-consuming tasks	-Abstraction of computing resources -Full control of the features and tools -Seamless platform updates	develop and customize your application without worrying about the infrastructure
IaaS Infrastructure as a Service	Provides users with the cloud computing infrastructure they need to perform generalized or specialized tasks	Dynamic scaling save money by only paying for what you are actively using	provides the most flexibility as virtualized, cloud-based computing resources & not software

Top Challenges in Big Data Ecosystem

- The biggest challenges with data today
    - **Data Quality**
    - **Staleness**
- According to Gartner, the main challenges are
    - Data Silos
    - Fragmented Tools
    - People with skill set to wield them
- This results in
    - Inaccurate Insights
    - Delayed & hence unusable results



Best Practices for Big Data Platforms

- Build Decoupled Systems
  - Storage & compute
  - Service oriented
  - Leverage Cloud storage in open format
- Right tool for the right job
  - Multiple use cases leveraging the same data with different tools
  - Consider Trade offs: Latency, throughput, Access Patterns
- Log centric design patterns in a multi tenant setup
  - Immutable logs - so that the sequence of changes can be viewed/replayed
  - Multiple views of the data depending on consumer needs ex. PII data masking
- Cost to build
  - Speed to Insights guides Build Vs Buy
  - There is a cost to build (time) and a cost to buy ($)
    - in-house expertise is leveraged at the cost of time

Business Justification for Tech Spending

- Tech should aid value and growth rather than be viewed as a cost allocation.
  - So it is important to <u>demonstrate value of tech investment</u>
- A joint business-technology strategy
  - Helps clarify the role of technology in driving business value
  - Provide a transformation agenda that can inform the organization’s tech investment strategy.
- Financial metrics
  - Including value growth, market share, ROI, earnings per share, profitability, margins, and revenue
  - Depending on business context and industry and market conditions.
  - Informed investment decisions likely require an understanding of technology’s impact on these key performance indicators (KPIs).
  - *Every technology investment should have a calculated and preferably tangible return.*
- Balance Infrastructure gains with Productivity and Capability gains
  - Consider CAPEX Vs OPEX
  - Risk Assessment and backup plans
  - Tunable cost platforms
  - Data is an asset, has to be governed and protected from inappropriate access/breach

Map Tech Solutions to Business needs to ensure successful implementation of Data Platforms

	Technology	Business
Present	Current Technology Challenges	Negative Business Consequences
Future	Proposed Technology Changes	Positive Business Outcomes

Demand Mapping with a small POC (6 month, 1 year, 3 year)		
Present	Project with current cost	As -Is - understand all the cost components of existing system
Future	Project with new cost	Additional Capex, Use Cases, Training Some time with 2 systems in parallel

Qualifier	
M	Metric
E	Economic Buyer
D	Decision Criteria
D	Decision Process
P	Partners
I	Identified Pain
C	Champion
C	Competition

Homework

Simplifying Data Engineering - first 3 chapters

Spark - The Definitive guide

- Use as Reference, Read Introduction to Spark & Data Frames

Assignment - 1 (due Mon, Sep 20 at midnight)

- Individual submissions
- Export the completed notebook in .dbc or .html format
- Upload the completed notebook to Canvas

Lab #0

- Introduction to the Databricks Platform using the Free Edition
  - Clusters & Notebooks & default datasets
  - Execute code in multiple languages (magic commands)
  - Read and Write data using csv, json
  - Spark Dataframe
  - Create database and table
  - Query table and plot results
  - Add notebook parameters with widgets
- Use Case: Asset Valuation
  - Industry: Real Estate
  - Evaluate neighborhoods to predict house price using linear regression