Introduction to Big data

Vimala Nunavath, Ph.D

Vimala.Nunavath@uia.no

Outline

- “Hangovers” (from Session 1)
  - the essays
  - the programming projects
  - introduction to EM
- Introduction to big data
- Chapter presentations
  – learning to read and present scholarly work
- Practical Session: introduction to Apache Spark

Individual essay

- The essay shall present and discuss selected theory, technology and tools related to big data technologies and EM, backed by scholarly and other references
  - – counts 30% of final grade
  - – presentations: December 5th
  - – deadline: December 4rth 1400
  - - Optional deadline: send me a brief informal email proposal by Monday September 9th 1400
  - - Final deadline: by Friday September 13th, 1400
  - – suggested length is 4000-6000 words.
- Encouraged:
  - – a scientific paper

Some possible Essay Themes:

- Types of Big Data challenges and analytical methods in terms of disaster management: A systematic literature review
- Exploring different visualization methods for social media big data: A systematic literature review
- Exploring different machine learning approaches for natural disaster management: A systematic literature review
- Evaluating different machine learning approaches for man-made disaster management
- Social media analytics for natural disaster management
- The Rising Role of Big Data Analytics and IoT in Disaster Management: Recent Advances, Taxonomy and Prospects
- How social media enhances emergency situation awareness?
- Discovering Big data Technologies for natural disaster management: recent research and future directions
- Discovering Big data Technologies for man-made disaster management: recent research and future directions
- Internet of Things (IoT) Considerations, Requirements, and Architectures for Disaster Management System
- Exploring IoT Applications for Disaster Management: recent research and future directions

Group programming project

- The project shall develop an application that can be used for emergency management. Development and run-time platform is free choice, as is programming language. The project should be carried out in groups of three and not more. Working individually or in pairs is not recommended.
- Counts 40% of final grade.
- Final presentation: Friday December 6th
- Submission deadline: Friday 13th, December 1400
- Topics in the third session

Outline for Introduction to Big data:

- What is big data and its characteristics
- Enablers of the big data
- Big data sources
- Existing big data technologies

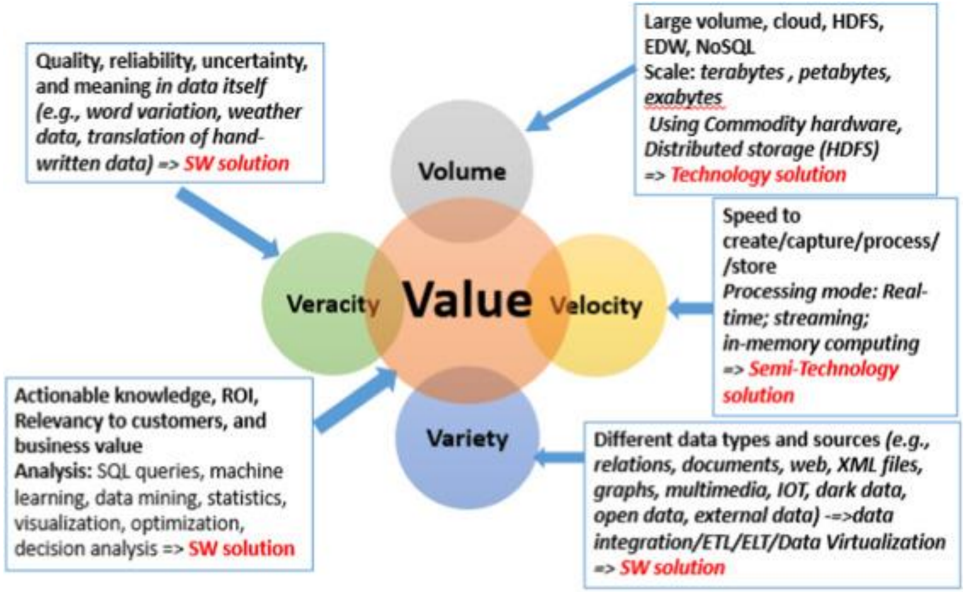
Big Data:

- Popular since late 2000's
  - buzzword, over-hyped, maybe already waning
  - but there is a (disruptive) reality behind it:
    - ever increasing amounts of available data
    - go beyond capabilities/capacities of established computing techniques and tools
    - calls for new understandings, techniques and tools
- Our working definition for now:

  "the ever-increasing amount of available data today that go beyond the capabilities/capacities of existing solutions and thus calls for new understandings, techniques and tools".

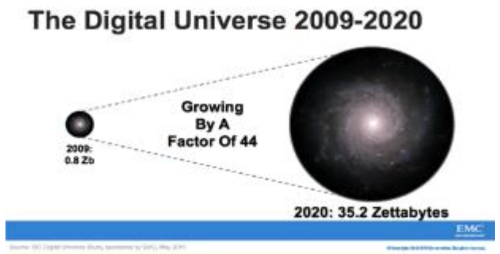
Characteristics of Big data:

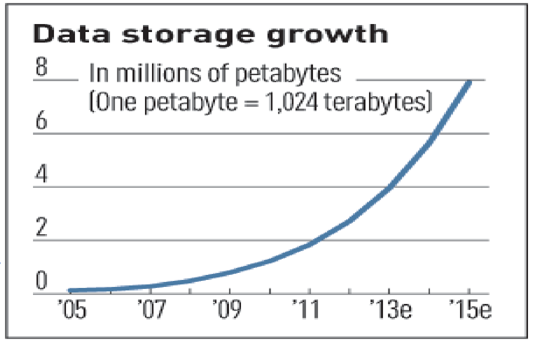
- The “three V's” (3V):
  - volume, velocity, variety – at once
    - old days: you could only have two of the three
  - also two more: veracity, value
- Other characteristics:
  - exhaustive in scope: “n = all”
  - fine-grained in resolution
  - indexical
  - relational in nature
  - flexible: extensional
  - flexible: scalable

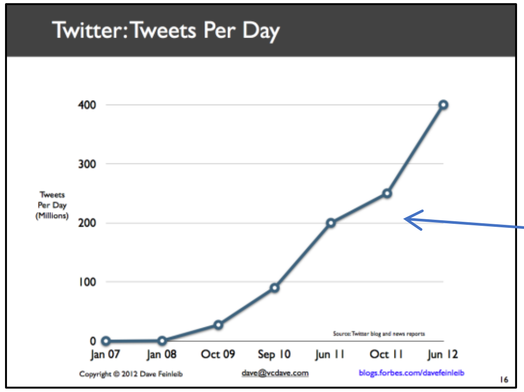


Volume:

- **Data Volume**
  - 44x increase from 2009 2020
  - From 0.8 zettabytes to 35zb
- Data volume is increasing exponentially







Exponential increase in collected/generated data

Measurements of digital data

Unit	Size	What it means
Bit (b)	1 or 0	Short for 'binary digit', after the binary code (1 or 0) computers use to store and process data
Byte (B)	8 bits	Enough information to create an English letter or number in computer code
Kilobyte (KB)	1,000, or 2^{10} bytes	From 'thousand' in Greek. One page of typed text is 2KB
Megabyte (MB)	1,000KB; 2^{20} bytes	From 'large' in Greek. The complete works of Shakespeare total 5MB. A typical pop song is about 4MB
Gigabyte (GB)	1,000MB; 2^{30} bytes	From 'giant' in Greek. A two-hour film can be compressed into 1–2GB
Terabyte (TB)	1,000GB; 2^{40} bytes	From 'monster' in Greek. All of the catalogued books in America's Library of Congress total 15TB
Petabyte (PB)	1,000TB; 2^{50} bytes	All the letters delivered by America's postal service in 2010 amounted to around 5PB of data
Exabyte (EB)	1,000PB; 2^{60} bytes	Equivalent to 10 billion copies of *The Economist*
Zettabyte (ZB)	1,000EB; 2^{70} bytes	The total amount of information in existence in 2010 was forecast to be around 1.2ZB
Yottabyte (YB)	1,000ZB; 2^{80} bytes	Currently too big to imagine
	The prefixes are set by an intergovernmental group, the International Bureau of Weights and Measures. Yotta and Zetta were added in 1991; terms for larger amounts have yet to be established.	

Source: *The Economist* (2010).

Velocity:

- Velocity:
  - created rapidly, in or near real time
  - analysis on the fly, not always storing it all

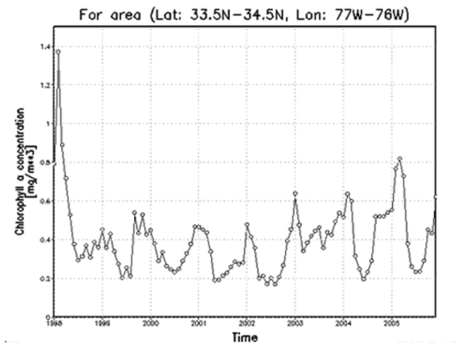
- Data is begin generated fast and need to be processed fast
- Online Data Analytics
- Late decisions ➔ missing opportunities
- **Examples**
  - **E-Promotions:** Based on your current location, your purchase history, what you like ➔ send promotions right now for store next to you

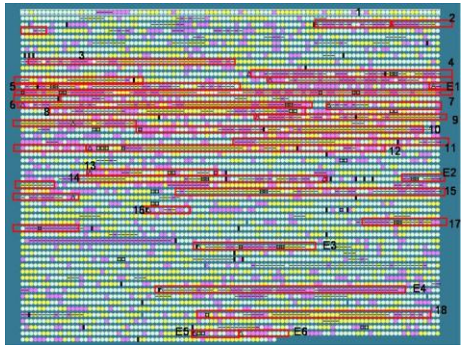
  - **Healthcare monitoring:** sensors monitoring your activities and body ➔ any abnormal measurements require immediate reaction

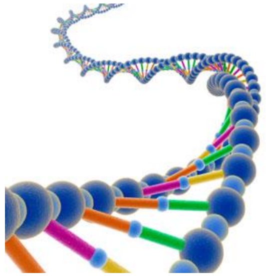
Variety:

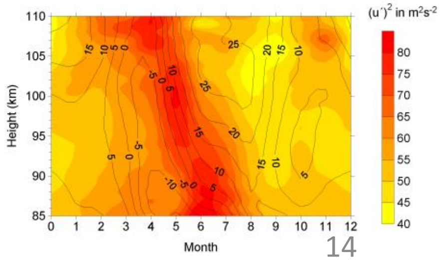
To extract knowledge → all these types of data need to linked together

- Structured, semi-structured and unstructured
  - new sources such as
    - natural language; microblog and other messages; social media conversations; sensor data; photos; video and sound recordings; PDFs/scans...
  - some temporal, some spatial, some both, some neither
  - some socially networked, some thematically grouped





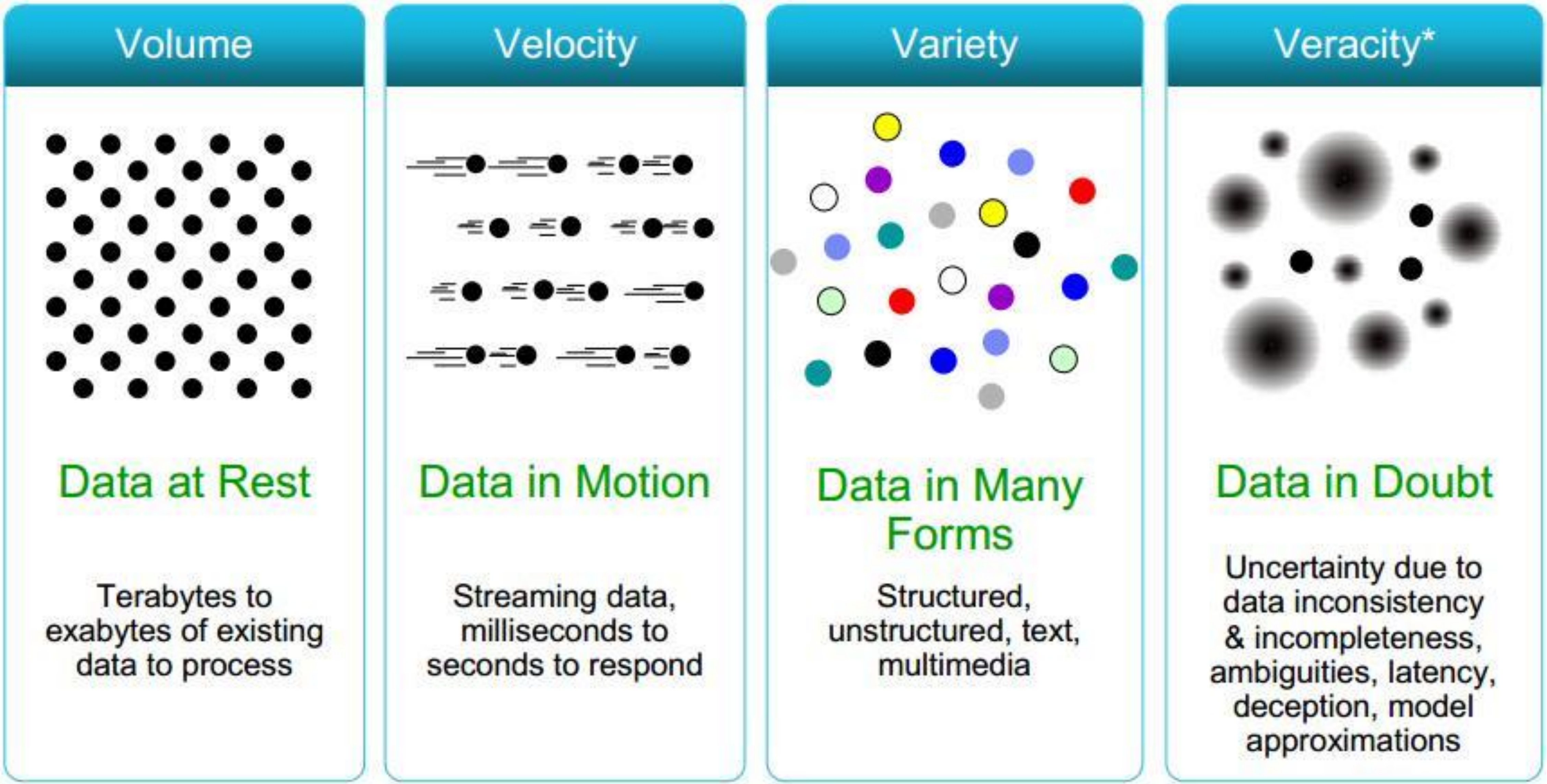




Veracity and Value:

- Veracity:
  - the trustworthiness of data: quality
    - accuracy, correctness, provenance (i.e., source of origin)
  - big data quality is uneven and can be low
    - e.g., microblog streams
  - how and when can volume make up for quality?
- Value:
  - how to make value out of the data?
    - both commercial and societal
  - e.g.: understand/serve customers/citizens; optimize business processes; "nowcasting"; assess teaching effectiveness; societal safety; detect cyber crime...

Some make it 4V's



Exhaustiveness, resolution, indexicality:

- Exhaustiveness
  - capturing and analyzing data about everyone/-thing
    - instead of sampling
- Fine-grainedness in resolution
  - aiming to be as fine-grained as possible
  - collecting, storing and analyzing smallest data points
    - instead of storing aggregate values
- Indexicality
  - unique identifiers for everyone and everything
  - trying to match different identifiers for the same person or thing (e.g., user names/handles)
  - using IRIs to identify resources on the Web of Data

Relationality:

- People and things are described in ways that make them combinable with
  - other related persons and things
  - other descriptions of the same persons and things

Flexibility:

- Extensionality
  - easy to add new data to the data set

- Scalability:
  - big datasets should be able to scale rapidly
  - use of grid computing, cloud servers, NOSQL databases (Not-Only SQL)

Enablers of big data:

- Computation
  - “Moore's law” of transistor numbers (1965 –)
- Networking
  - “Gilder's law” of network bandwidth (2000 –): global bandwith doubles every 6 months
- Storage (cloud, *aaS, NOSQL)
- Pervasive and ubiquitous computing
  - sensors and actuators
  - from dumb to smart things (cars)
  - exhaustive data collection
  - “ambient computing”, “the age of everyware”

Pervasive versus ubiquitous computing:

- Pervasive computing:
  - computing “in everything”
  - make them interactive and smart
  - divergent: more and more things become smart
  - needs situational awareness
- Ubiquitous computing:
  - computing “in every place”
  - moves with the person
  - convergent: smart things we carry do more and more tasks
  - needs context and location awareness

Enablers of big data:

- Indexicality
  - growth of unique identifiers
  - people: user names/handles, personal numbers / SSNs, passports, driver's licenses, health cards, biometry, IMSIs
  - things (and information): product type codes, RFID for individual products, auto passes, MAC addresses, IMEIs, IRIs, including ISBNs, ISSNs, DOIs, etc.
  - places: post codes, addresses, geo coordinates
- Machine-readable identification
  - more and more are becoming digital
- ...and remote readable

Sources of big data:

- Three types:
  - directed
  - automated
  - volunteered
- Directed data collection
  - organized and structured surveillance
  - personal or through technological lens
  - census, government forms, inspections, CCTV cams
  - surveillance technology is becoming digital, smarter, directable, internetworked...

Automated data collection:

- Automated surveillance
  - e.g., smart electricity meters, electronic transportation tickets, passenger counting systems, car tolls, radar/LiDAR speed guns, ANPR
- Digital devices
  - smart phones/tablets + lots of others
  - actively produce data
  - primary: cameras, videos, GPS units, medical devices
  - exhaust: mobile phones (also primary), cable boxes
  - logjects = objects that log their (+ their users') history
  - objects can also be logged by others • e.g., mobile-device triangulation

Automated data collection:

- Interaction data
  - all ICT-based transactions leave traces
  - using a web shop, net bank, ATM
  - sending an email
  - accessing the internet from home or a mobile device
- Scan data
  - machine-readable identification codes
  - barcodes, QR (“Quick Response”) codes
  - magnetic cards, chip card/smart card/ICC
- Sensor (sensed) data
  - inexpensive sensor generate continuous data streams
  - smart cities gauging noise, temperature, light, CO2 ...

Volunteered data collection:

- Social media, collective projects (online)
    - production + consumption = prosumption
- Transactions
    - voluntary registration, clickstreams, review data
- Some of the automated collection was volunteered:
    - actively produced data
    - primary: cameras, videos, GPS units, medical devices
    - some logjects (objects that log history)
- Sousveillance
    - (fr.) sur-: above, sous-: below
    - self-monitoring, e.g., wearable fitness equipment, dieting apps

Volunteered data collection:

- Crowdsourcing
  - to create one new product
  - to create many new products/concepts/ideas
  - to assess many existing products/concepts/ideas
- Citizen science

  - ‘communities or networks of citizens … act as observers in some domain of science

Big Data as a Disruption:

- Disruptive technology:
  - a technology that displaces established ones, and shakes up existing or creates new industries
  - e.g., PCs, the internet, digital media, social media
- Big data is disruptive
  - it creates new data-driven organization forms
  - new ways of doing research and science
  - new ways of creating and maintaining products and services
  - new threats to privacy and social order
- ...too easy to shrug off (just) as a hype/buzzword

Data-driven organizations:

- “The next phase of the knowledge economy, reshaping the mode of production” (RK, p. 16)
  - **inward:** monitor, evaluate performance in real time; reduce waste and fraud; improve strategy, planning and decision making
  - **outward:** design new commodities, identify and target new markets, implement dynamic pricing, realize untapped potential, gain competitive advantage
- **Goals:** run more intelligently; flexibility and innovation; reduced risk, cost, losses; improved customer exper., return on investment, profit

New ways of doing business:

- **Marts (Walmart, Kohl's):** analyze sales, pricing, economic, demographic and weather data to tailor local product selection and price markdowns
- **Online dating:** sift through personal characteristics, reactions and communications to improve matches
- **NY Police:** analyze data on past arrests, paydays, sporting events, weather and holidays to deploy officers optimally
- **Professional sports:** massaging sports statistics to spot undervalued players
- **Education:** analyze data from learning management systems to improve teaching / studying

(from: Steve Lohr (2012): The Age of Big Data, NYTimes.com)

Big data Technologies:

- Big Data Technology: “as a Software-Utility that is designed to **Analyse**, **Process** and **Extract** the information from an **extremely complex and large data sets** which the Traditional Data Processing Software could never deal with”.

- Types of Big Data Technologies:

  - Operational Big Data Technologies

  - Analytical Big Data Technologies

Big data Technologies:

- Types of Big Data Technologies:

  - Operational Big Data Technologies
    - The Operational Big Data is all about the normal day to day data that we generate. E.g., Online Transactions, Social Media, or the data from a Particular Organisation etc.
    - consider this to be a kind of raw data used to feed the Analytical Big Data Technologies.

  - Analytical Big Data Technologies
    - like the advanced version of Big Data Technologies.
    - Analytical big data is where the actual performance part comes into the picture and the crucial real-time business decisions are made by analyzing the Operational Big Data.
    - E.g. Stock marketing, Carrying out the Space missions where every single bit of information is crucial, Weather forecast information, Medical fields where a particular patients health status can be monitored.

Big data Technologies:

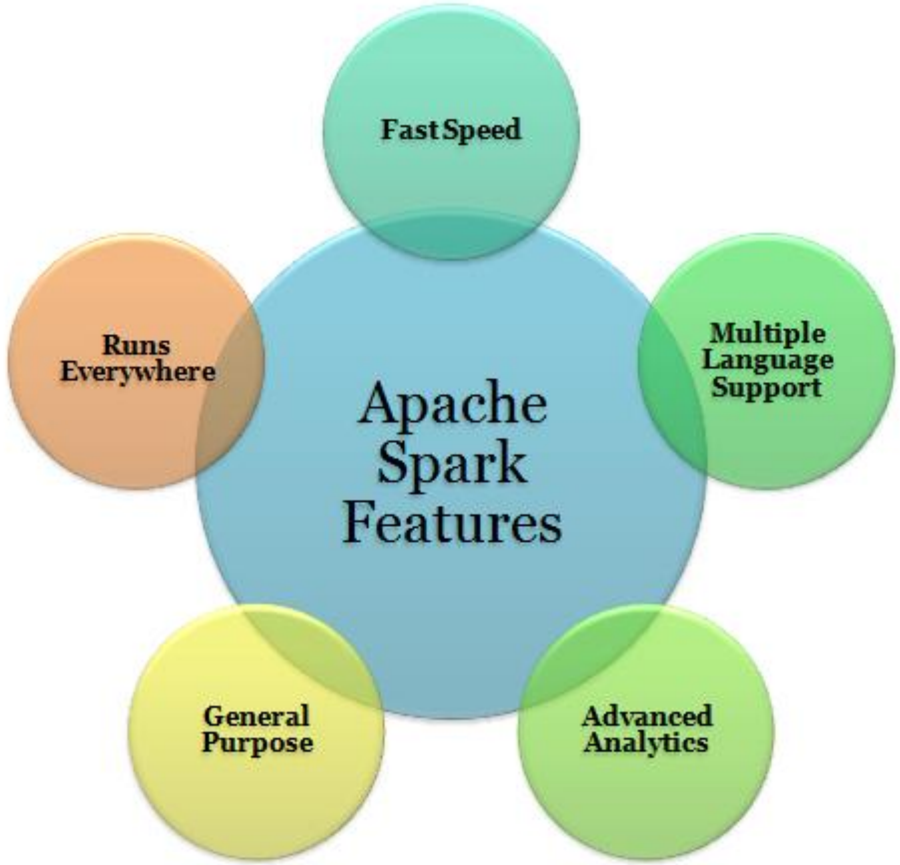
- Big data technologies are divided into 4 fields which are classified as follows:
  - Data Storage
  - Data Mining
  - Data Analytics
  - Data Visualization

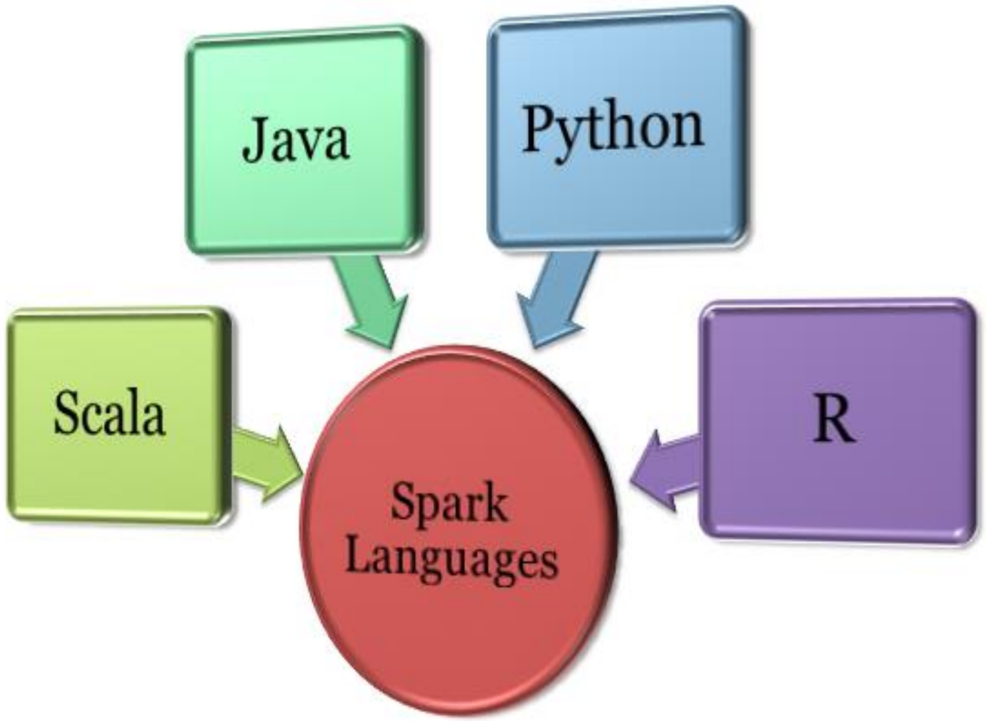
Big data Technologies:

Big data Technologies:

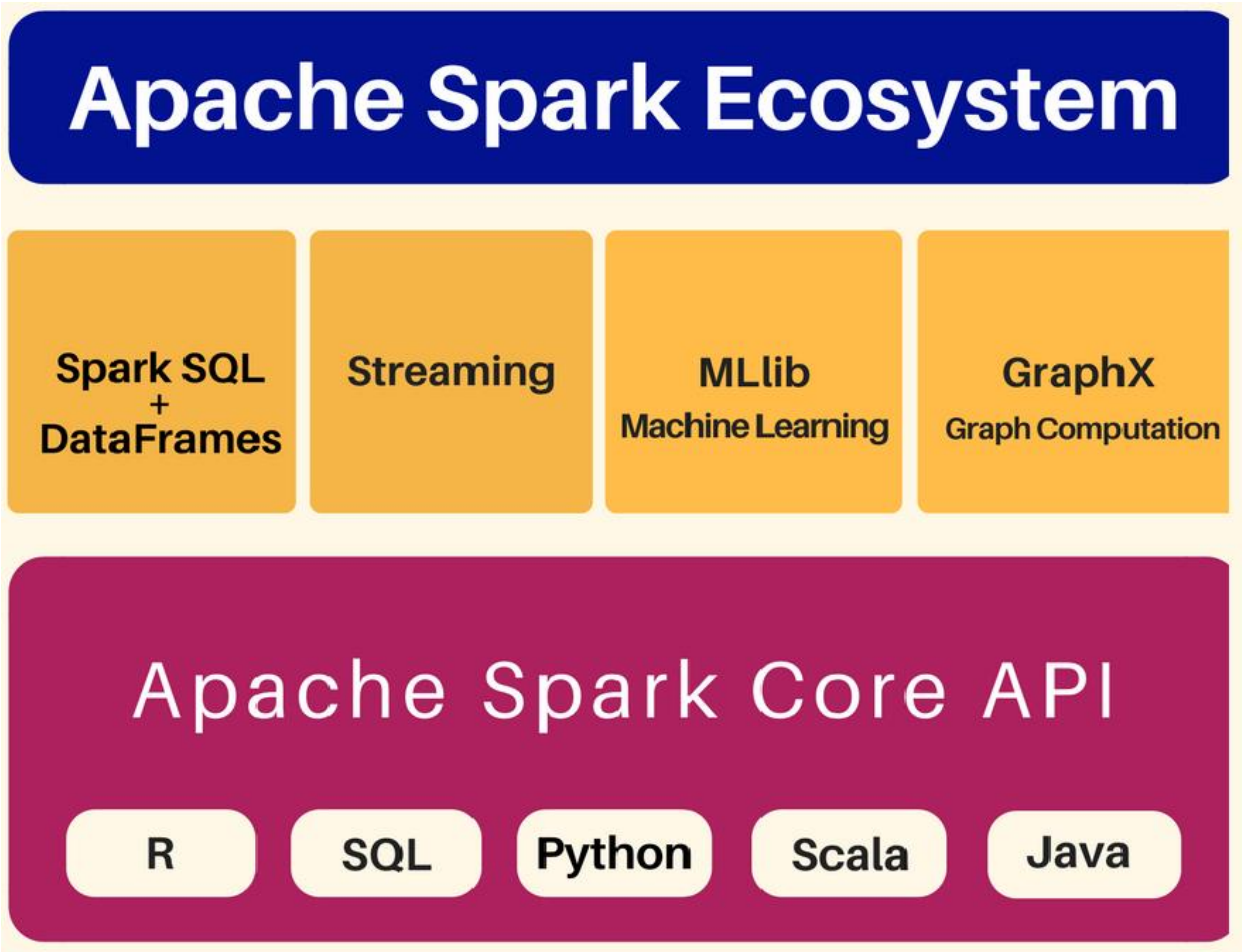
- **Apache Spark:**
  - Popularly known as “lightning fast cluster computing”.
  - an open-source framework for the processing of large datasets.
  - It is the most active Apache project of the present time.
  - it’s rapid success is due to its power and ease-of-use.
  - It is more productive and has faster runtime than other *BigData based analytics.*
  - written in Scala and provides APIs in Python, Scala, Java, and R.
  - important feature of Apache Spark is its in-memory cluster computing that is responsible to increase the speed of data processing.

Apache Spark Features





Components of Apache Spark Ecosystem



Spark Core

- The main execution engine of the Spark platform is known as Spark Core.
- All the working and functionality of Apache Spark depends on the Spark Core including memory management, task scheduling, fault recovery, and others.
- It enables in-memory processing and referencing of big data in the external storage systems.
- It is responsible to define RDD (Resilient Distributed Dataset) by an API that is the programming abstraction of Spark.

Spark SQL and DataFrames

- the main component of Spark that works with the structured data and supports structured data processing.
- Provides a programming abstraction called DataFrames.
- performs the query on data through SQL and HQL (Hive Query Language, Apache Hive version of SQL).
- This integration of SQL with advanced computing medium combines SQL with the complex analytics.

Spark Streaming

- It is responsible for the live stream data processing such as log files created by production web servers.
- It provides API for the manipulation of data streams, thus makes it easy to learn Apache Spark.
- It also helps to switch from one application to another that performs manipulation of real time as well as stored data.
- This component is also responsible for throughput, scalability, and fault tolerance as that of the Spark Core.
- It readily integrates with a wide variety of popular data sources, including HDFS, Flume, Kafka, and Twitter.

MLlib

- It is the in-built library of Spark that contains the functionality of Machine Learning, known as MLlib.
- It provides various ML algorithms such as clustering, classification, regression, collaborative filtering and supporting functionality.
- It is a scalable machine learning library that delivers both high-quality algorithms (e.g., multiple iterations to increase accuracy) and blazing speed (up to 100x faster than MapReduce).
- The library is usable in Java, Scala, and Python as part of Spark applications.

GraphX

- the library that enables graph computations.
- also provides an API to perform graph computation by allowing users generate directed graph using arbitrary properties of the edge and vertex.
- Along with the library for manipulating graphs, it provides many operators for the graph computation.

Why Apache Spark?

- Ease of use
- High-performance gains
- Advanced analytics
- Real-time data streaming
- Ease of deployment

Resilient distributed dataset (RDD):

- It is a fundamental data structure of Spark. Spark revolves around the concept of a *resilient distributed dataset* (RDD).
- It is an immutable distributed collection of objects.
- Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.
- There are two ways to create RDDs: *parallelizing* an existing collection in your driver program, or *referencing* a dataset in an external storage system, such as a shared filesystem, HDFS, HBase, or any data source offering a Hadoop Input Format.
- either transform data or take actions on that data.

Resilient distributed dataset (RDD):

- RDDs support two types of operations:
  - *Transformations:* create a new dataset from an existing one,
  - *Actions:* return a value to the driver program after running a computation on the dataset.

Transformation	Meaning
map(*func*)	Return a new distributed dataset formed by passing each element of the source through a function *func*.
filter(*func*)	Return a new dataset formed by selecting those elements of the source on which *func*returns true.
flatMap(*func*)	Similar to map, but each input item can be mapped to 0 or more output items (so *func* should return a Seq rather than a single item).

RDD Operations:

- Transformations:

Transformation	**Meaning**
map(*func*)	Return a new distributed dataset formed by passing each element of the source through a function *func*.
filter(*func*)	Return a new dataset formed by selecting those elements of the source on which *func*returns true.
flatMap(*func*)	Similar to map, but each input item can be mapped to 0 or more output items (so *func* should return a Seq rather than a single item).

RDD operations:

- Actions:

Action	**Meaning**
reduce(*func*)	Aggregate the elements of the dataset using a function *func* (which takes two arguments and returns one). The function should be commutative and associative so that it can be computed correctly in parallel.
collect()	Return all the elements of the dataset as an array at the driver program. This is usually useful after a filter or other operation that returns a sufficiently small subset of the data.
count()	Return the number of elements in the dataset.
first()	Return the first element of the dataset (similar to take(1)).
take(*n*)	Return an array with the first *n* elements of the dataset.

- *What to do in Two Weeks? ...and in the meantime :-)*