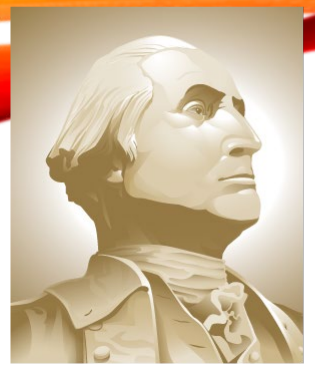


THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

INTRODUCTION TO BIG DATA AND ANALYTICS CSCI 6444

Prof. Roozbeh Haghnazar

Slides Credit:

Stephen H. Kaisler, D.Sc.and Prof. Roozbeh Haghnazar

Roozbeh Haghnazar

ROOZBEH@GWU.EDU

- Started Programming in1991 with Commodore 64
- Played several roles in technology, such as Developer, Modeler, Designer, Architect, Leader, CTO, etc.
- Teach Software Eng., Distributed Systems, Data Base Design Principles, Data Visualization, Operating System.
- Data Science Lead in NWITS-USGS
- Tech Lead in Spirent Communications

About This Course

- Be prepared! (course prerequisites)
  - Programming Background
  - Database Modeling
  - Having an overview of Data Management concept
- Be involved!
  - "Raise hand", ask questions, discuss, etc.
  - Asynchronous opportunities will be available
- Be ready to code!
  - You will need to use **Python** for your assignments
  - Mostly group projects

PURPOSE

- This course is an introductory course that will cover a lot of topics in Big Data and Analytics.
- The objectives are:
  - To introduce students to some of the concepts, issues and challenges in dealing with Big Data
  - To examine the types of analytics and work with a few of the tools to process some relatively Big Data sets.
  - To understand the type of advanced analytics beyond simple statistical analysis, data mining, and statistical machine learning which address complex problems facing business, society, science and engineering today.
  - To describe the roles of data scientist and analytic scientist
  - To practice some of the techniques through class projects.

RESOURCES

- Slack: (linked from website, join after class)
- GitHub for collecting assignments
- Blackboard for grades, class meetings, and office hours

- Visual Studio Code – recommended IDE
  - Live share plugin allows group collaboration / help in office hours

Semester Outline

- Building Blocks
  - Introduction to Big Data Management and Analysis
  - Scalable Execution: Processes, threads, parallelism vs concurrency
  - Apply the concepts on a real and practical projects
- **Principles** of Analysis
  - Data preparation: Cleaning, Standardization, etc
  - Descriptive and Predictive Analytics
  - Structured and Non-Structured Data
- Big Data in **Practice**
  - Cloud Computing
  - Web, Mobile, and IoT
  - Reliability: Replication and Fault Tolerance
  - Performance: Metrics and Modeling Large Scale Systems

4 programming assignments
Midterm ???
Large group project

TEXTBOOKS

Required:

Rampolla, M.L. 2014. A Short Guide to Writing in History, 9th Ed. St. Martin's Press 0312403577 or later

https://www.amazon.com/Pocket-Guide-Writing-History/dp/1319113028 - about $33 new, but there may be used versions.

Optional:

Data Science Books; Big Data Books (as PDFs), NoSQL books, etc.

Note: Rampolla's book contains the rules for writing good term papers. I have used it for over 25 years through various editions. When I grade your term papers for spelling, usage, structure, etc., I will be using Rampolla as the key reference. So, I strongly recommend that you buy this book and read it. There is NO excuse for a poorly written paper at the Upper Undergraduate/Graduate Level!!

Class Notes

- There will be class handouts provided every week. These will be posted to the class Blackboard/Github site.
  - The class notes will serve as an outline, but I will provide additional material during the lectures.
- The final exam will be drawn from the class notes. It will be on Blackboard.
- It is incumbent upon you to attend every lecture (unless you are ill or on travel for your job) or to obtain class notes from someone who was virtually present (and, hopefully, awake and not surfing the net).
- *The additional books on Blackboard are supplemental to the lectures. You should read for absorption and understanding, not for regurgitation.*
- ***Because this course is a mix of graduate and undergraduate students, the final exam date will be determined by the registrar for undergraduate students.***

TERM PAPERS/PROJECT

Term Papers

- If English is NOT your primary language (or, even if it is), I encourage you to make use of the GW Writing Center, http://www.gwu.edu/~gwriter/. You should be able to contact them online for assistance and webex to talk to them.
- I have posted the term paper requirements on Blackboard. Read them and adhere to them. I grade strictly to these requirements.
- Submit to me two short paragraphs on the topic you choose for your term paper by lecture 3.
- Begin your term paper early – time flies quickly!
- You must use Rampolla as your guide. Points are subtracted for errors where guidance was provided by Rampolla.

Use of ChatGPT and AI in Preparing Term Papers

- The Provost's Office is examining student use of artificial intelligence (AI) programs, such as ChatGPT, to complete coursework. This is a complicated set of issues that has evolved rapidly since ChatGPT's launch in November. It is important to appreciate how these programs can be used to advance learning objectives, but also how the Code of Academic Integrity is implicated by the use of these programs. At present, ChatGPT and similar resources have been added to the Guide to Clarifying Academic Integrity Expectations, a model that faculty may use to indicate which resources are permitted or prohibited on various assessments. The Provost's Office will issue additional guidance soon on this matter as well.
- Read the Guide at https://studentconduct.gwu.edu/promoting-academic-integrity

Resource	Permitted	Permitted with Citation	Prohibited
ChatGPT or other artificial inteligence.			X
Chegg, Course Hero, Quizlet, and similar sites focused on academic assessments.			X
Classmates in your assigned group.		X	
Classmates, including via GroupMe or other shared conversations		X	
Classmates in other groups, not your own.		X	
Course materials on Blackboard.		X	
Course materials not on Blackboard.		X	
Gelman Library Research Services		X	
Google translate, other translation services and tools, or other tools of "artificial intelligence" (broadly interpreted). [Faculty may wish to specify further.]		X	

GW Writing Center			
Material from outside of this course (e.g., library books, notes from other courses, online material, Wikipedia, YouTube videos, etc).		X	
Material from students formerly enrolled in the course (when used without permission, this may result in academic integrity violations for all students involved).		X	
Notes page designated for this purpose (e.g., you may bring one page of notes to an in-class exam).	Not Applicable		
Notes taken in course meetings (including office hour meetings).	X		
Other people (not classmates as noted above).		X	
Recorded lectures (from this class, if recording was done or permitted by instructor).		X	
Recorded lectures, talks, podcasts, videos (from a source other than this class).		X	
A tutor (from GW's Academic Commons or elsewhere at GW).		X	
A tutor not affiliated with a GW service.		X	
All other resources not specified, unless you receive direction otherwise from the course leaders.			X

CLASS PROJECTS - I

What we have to learn to do, we learn by doing.

— Aristotle, *Ethica Nicomachea II* (ca. 325 B.C)

- The bulk of the grade for this course will consist of several class projects using Python.
- Working in teams of 2/3 people, you will learn how to process big data sets from several analytic perspectives.
- I will assign the teams arbitrarily before the first class and set up groups in Blackboard/Github.
- The team will be graded on their group results. Therefore, it is up to the team to make sure everyone does his or her fair share.
- Do NOT ask me to resolve team conflicts or referee team disputes. As undergraduate/graduate students, you will need to get along for a semester in order to achieve a common goal.

Class Projects - II

- The class projects will require substantial time outside of the class room.
- ***Feel free to use <u>all</u> available hours not in class itself or working on the term paper to work on the class projects. <u>Sleeping and eating are optional</u>!!***
- The class projects are described in separate handouts.

<u>Why these projects?</u>

- I have taught computer science for 15 years. It is my experience that, today, most computer science students do not really understand the material without hands-on experience.
- I feel as a Computer Science Undergraduate/Graduate Student you should be able to demonstrate not only mastery of the conceptual material but the practical application of that material to real problems.

Introduction to Big Data

BIG DATA

- Why do we need this amount of infrastructure?
  - **Airbus A350**
    - Contains around 6000 sensors across the entire plane that generate 2.5TB of Data per day
  - **Airbus A380-800**
    - An Airbus A380-800 is equipped with up to 25,000 sensors to gather data for maintenance, operational monitoring, flight control, and other functions.
  - **Google:**
    - Google **doesn't publish an official daily search count** today; reputable industry estimates put it at **~8.5 billion searches/day** (2024). Some newer analyses suggest a higher range (≈**13–16 billion/day**), but the 8.5 B estimate is the conservative, commonly cited figure.
  - **Facebook:**
    - **Monthly Active Users (MAU): ~3.06–3.07 billion** in 2024–2025, per Meta-based roundups (use 3.07 B on slides).
    - **Daily data volume:** historical engineering posts put Facebook's data warehouse at **~300 PB stored** with **~600 TB/day** ingested (2014), and Meta Research has cited **~4 PB/day generated** (also historical). Meta doesn't publish a current daily data figure; use these as **scale markers, not 2025 numbers**.

And then BIG Data

- **Twitter** generates over 500 million tweets per day. Each tweet, retweet, like, and interaction further adds to the data pool.
- **Internet of Things (IoT):** It's estimated that 127 new IoT devices connect to the internet every second. These devices continuously generate data, from smart thermostats to connected vehicles.
- **Autonomous Vehicles**: Self-driving cars, while still in the developmental phase in many aspects, can generate up to 4 terabytes of data per day, including sensor data, video feeds, and system performance metrics.
- **Healthcare:** A single patient can generate up to 80 megabytes of data annually in imaging and electronic medical record data.

KB	Kilo Byte	1 thousand bytes
MB	Mega Byte	1 million bytes
GB	Giga Byte	1 billion bytes
TB	Tera Byte	1 trillion bytes
PB	Peta Byte	1 quadrillion bytes
EB	Exa Byte	1 quintillion bytes

EVERYBODY LIES

BIG DATA, NEW DATA, and What the Internet can Tell Us about Who we really are

Seth Stephens-Davidowitz

Big Data: The Digital Truth Serum

- Traditional surveys & interviews → people lie or hide.
- Google searches & online data → reveal what people *really* think, feel, and do.
- Big Data helps us uncover **hidden truths** about society, health, and behavior.
- Key message: ***"Everybody lies, but the data doesn't."***

Example 1: The American Dream

•**Myth:** “The American Dream is the same everywhere in the U.S.”

•**Big-data finding:** Upward mobility **varies dramatically by ZIP code**. Overall U.S. mobility lags top countries, **but** the best U.S. regions **match** the world’s leaders. Policy + local conditions (segregation, inequality, school quality, social capital, family structure) drive the gaps.

EXAMPLE 2: HEALTH & SUICIDE RISK

- **Problem:** Official suicide statistics are delayed & underreported.

- **Big Data Insight:** Millions of annual Google searches for phrases like *"how to kill myself"* strongly correlate with actual suicide rates.

- **Impact:** Real-time monitoring could guide mental health interventions and resource allocation.

- **Lesson:** Big Data can save lives by revealing crises faster than traditional methods.

Big Data: Handle with Care

- **Opportunities:**
  - Predict trends, improve policies, save lives.
  - Make social science more scientific.

- **Risks:**
  - Correlation ≠ causation.
  - Easy to be misled by spurious patterns.
  - Privacy concerns & potential misuse by corporations/governments.

- **Takeaway:** Big Data is powerful but must be used responsibly.

Big Data: Handle with Care

- **Curse of dimensionality:** With thousands of signals, some will "predict" by luck. Use strong out-of-sample tests, cross-validation, and regularization. Keep a truly blind holdout.

- **Overemphasis on what's measurable:** Metrics can hijack behavior (Goodhart's Law). Design metrics to reflect outcomes you actually want, not just what's easy to count.

- **Empowered corporations:** Predictive models can morph into opaque filters (loans, hiring) or aggressive **price discrimination**.

- **Empowered governments:** Area-level signals can guide resources, but targeting **individuals** from behavioral traces threatens civil liberties.

Big Data: Handle with Care

- **Methodological guardrails**
  - Pre-register hypotheses when feasible; separate **discovery** vs **confirmation**.
  - Holdout discipline: keep a **sealed test** untouched until the end; avoid iterative peeking.
  - Control false discoveries: multiple-testing corrections; shrinkage/regularization; parsimonious features.
  - Robustness: sensitivity analyses, placebo tests, falsification checks; test counterfactual stories.
  - Causality: prefer experiments / natural experiments; avoid causal claims from correlation-only patterns.

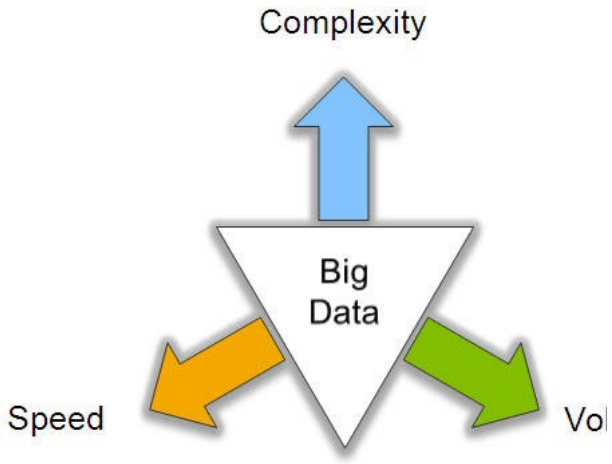
- **Governance guardrails**
  - **Purpose limitation** and data minimization; retain only what is necessary.
  - **Fairness**: evaluate group-wise error rates (FPR/FNR), calibration, and disparate impact; document trade-offs.
  - **Privacy**: de-identify; consider DP/k-anonymity; secure pipelines and access logs.
  - **Transparency**: model cards, datasheets; record features that are proxies for protected traits.
  - **Red-team** models for exploitation (e.g., predatory pricing, addiction loops).

What's Big Data?

No single definition; here is from Wikipedia:

- **Big data** is the term for a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.
- The challenges include capture, curation, storage, search, sharing, transfer, analysis, and visualization.
- The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, determine quality of research, prevent diseases, link legal citations, combat crime, and determine real-time roadway traffic conditions."

BIG DATA: 3V'S



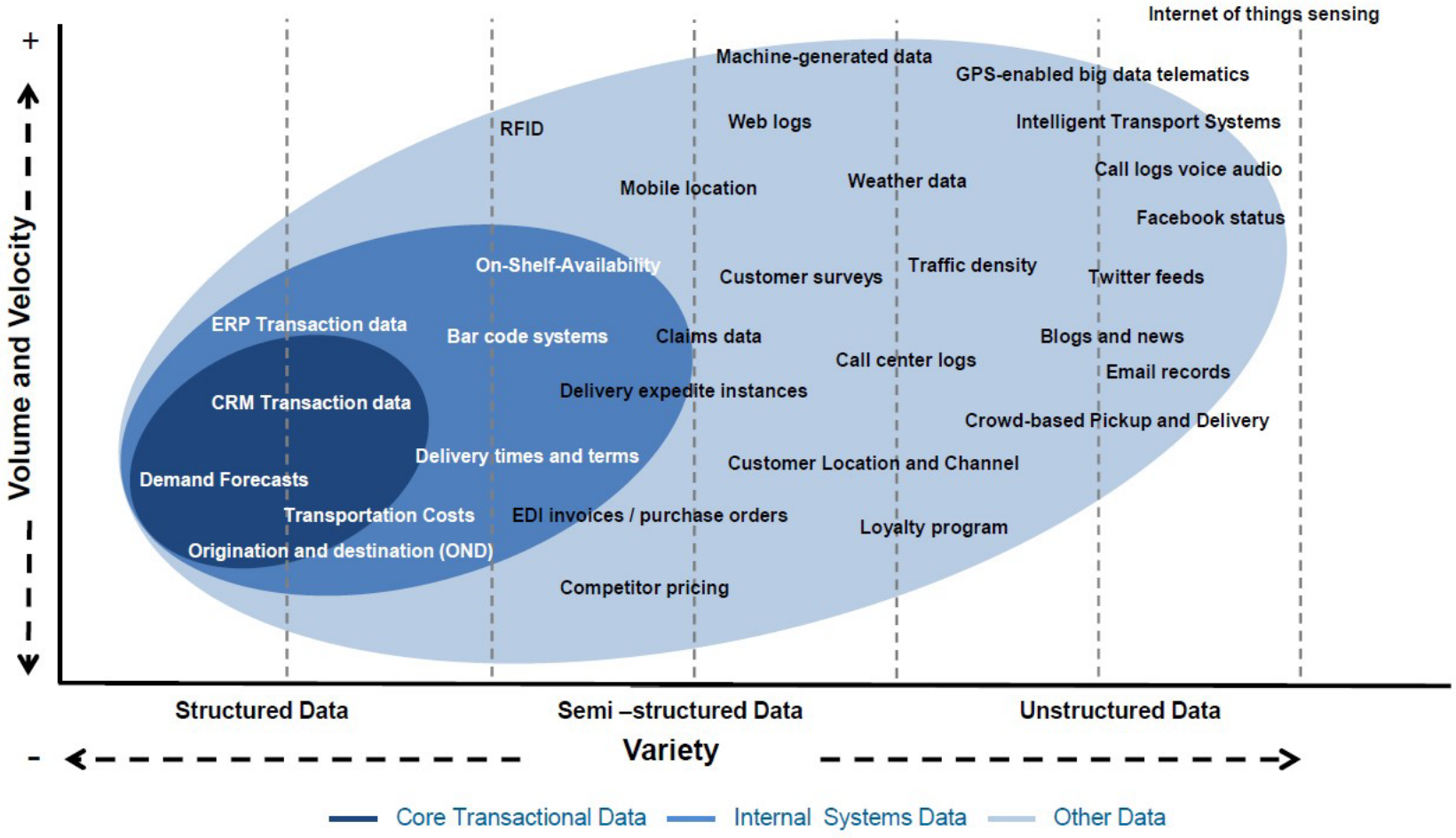
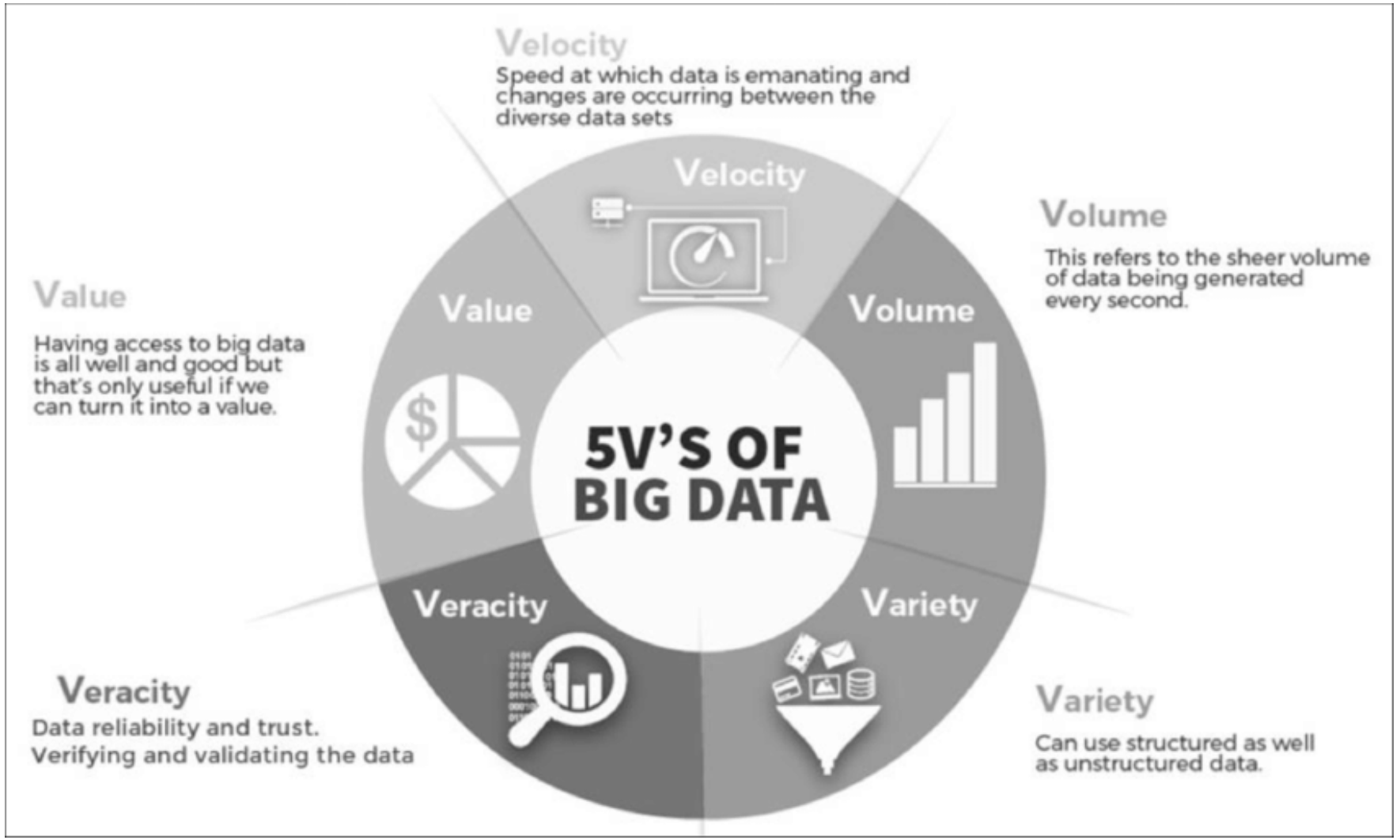


Figure 1. SCM Data Volume and Velocity vs. Variety

Big Data 5Vs


Velocity
Speed at which data is emanating and changes are occurring between the diverse data sets
Velocity
Volume
This refers to the sheer volume of data being generated every second.
Volume
Value
Having access to big data is all well and good but that's only useful if we can turn it into a value.
Value
5V'S OF BIG DATA
Veracity
Data reliability and trust. Verifying and validating the data
Veracity
Variety
Can use structured as well as unstructured data.
Variety

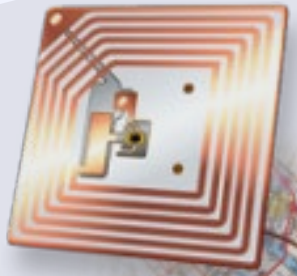
THE SCALE

12+ TBs **of tweet data every day**

2 TBs of data every day

25+ TBs **of log data every day**

30 billion RFID tags today (1.3B in 2005)



4.6 billion camera phones world wide

100s of millions of GPS enabled devices sold annually

76 million smart meters in 2009… 200M by 2014

2+ billion people on the Web by end 2011

Cloud Computing and Big Data

- One of the vital issues that organizations face with the **storage and management** of Big Data is the **huge amount of investment** to get the required hardware setup and software packages.
- Some of these resources may be **overutilized or underutilised with varying requirements overtime**. We can overcome these challenges by providing a set of computing resources that can be shared through cloud computing.
- Cloud platforms provide a **shared infrastructure** and **pay-as-you-go model** that lowers cost. This has helped enable the "big data revolution"

In-Memory Technology for Big Data

- Disks are really really slow! This might prevent us from analysing data in **real-time**
- **In-memory** big data computing tools overcome this issue by keeping the important data in memory.
  - But this isn't always possible! **Why not?**

What types of applications or analytics are a good fit for in-memory processing?

In-Memory Technology for Big Data

- **Real-Time Analytics:**
  - **Stream processing:** Real-time analysis of data streams, such as social media feeds, stock prices, or sensor data.
  - **Fraud detection:** Spotting unusual patterns or anomalies in transaction data as they occur.
  - **Monitoring and alerting:** Tracking system health, performance metrics, and triggering alerts based on predefined conditions.

- **Interactive Business Intelligence (BI) and Reporting:**
  - **Ad-hoc queries:** Allowing users to interactively explore datasets and generate reports without waiting for extended periods.
  - **Dashboards:** Real-time visualization and monitoring of key performance indicators (KPIs).
- **Iterative Machine Learning and Data Mining:**
  - **Recommendation systems:** Using iterative algorithms like matrix factorization, which benefit from fast data access.
  - **Clustering and classification:** Repeatedly accessing data for techniques like k-means clustering or building decision trees.

In-Memory Technology for Big Data

- **Graph Processing:**
  - **Social network analysis:** Analyzing relationships, communities, and trends in graph-based datasets.
  - **Shortest path analysis:** Finding the quickest routes in transportation networks or logistics.
- **Simulation and Modeling:**
  - **Financial simulations:** Running Monte Carlo simulations for risk assessment.
  - **Scientific simulations:** Conducting iterative experiments or modeling in fields like physics, biology, or meteorology.
- **Cache for Web Applications:**
  - **User session storage:** Storing active user session data for fast retrieval.
  - **Content delivery:** Caching frequently accessed website elements, such as images, CSS, or JavaScript files.
- **Batch Processing:**
  - **ETL processes:** Extracting, transforming, and loading data more quickly by leveraging in-memory processing.
  - **Data preparation:** Performing operations like filtering, sorting, or aggregation for subsequent analysis.
- **High-Performance Databases:**
  - **In-memory databases:** Examples include Redis, SAP HANA, and MemSQL, which are optimized for in-memory operations and can deliver high-throughput and low-latency transactions.
  - **Temporal data storage:** Retaining frequently accessed or recently inserted records in memory for rapid querying.
- **Event-Driven Applications:**
  - **Complex event processing (CEP):** Monitoring, analyzing, and acting on event patterns in real-time.

Big Data Techniques

- To analyze the datasets, there are many techniques available, some of which are as follows:
  - Massive Parallelism
  - Data Distribution
  - High-Performance Computing
  - Task and Thread Management
  - Data Mining and Analytics
  - Data Retrieval
  - Machine Learning
  - Data Visualisation

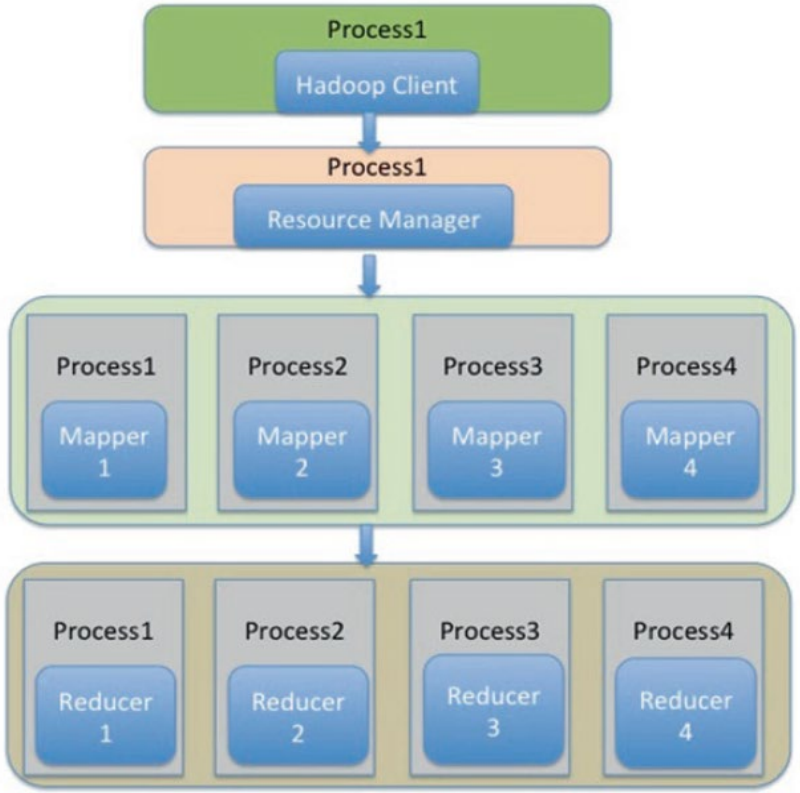
Fundamental Concepts of Distributed Computing Used in Big Data Analytics

Big Data and…

- **Heterogeneity**
- **Openness**
- **Security**
- **Failure Handling**
- **Concurrency**
- **Quality of Service**
- **Scalability**
- **Transparency**

MAP REDUCE / HADOOP

- The first popular "big data" platform for the cloud
- Can install **Apache Hadoop** on your own **IaaS** VMs/containers
- Or use a **PaaS** version:
  - AWS Elastic Map Reduce, Google Dataproc, Azure HDInsight
- MapReduce uses distributed systems concepts to achieve high scalability
  - Partitioning, failure detection, data replication for performance and reliability…
- MapReduce relies on a cloud infrastructure to provide its computational and storage resources



Multiprocessing model in the Hadoop runtime environment

MAP REDUCE **QUALITY OF SERVICE**

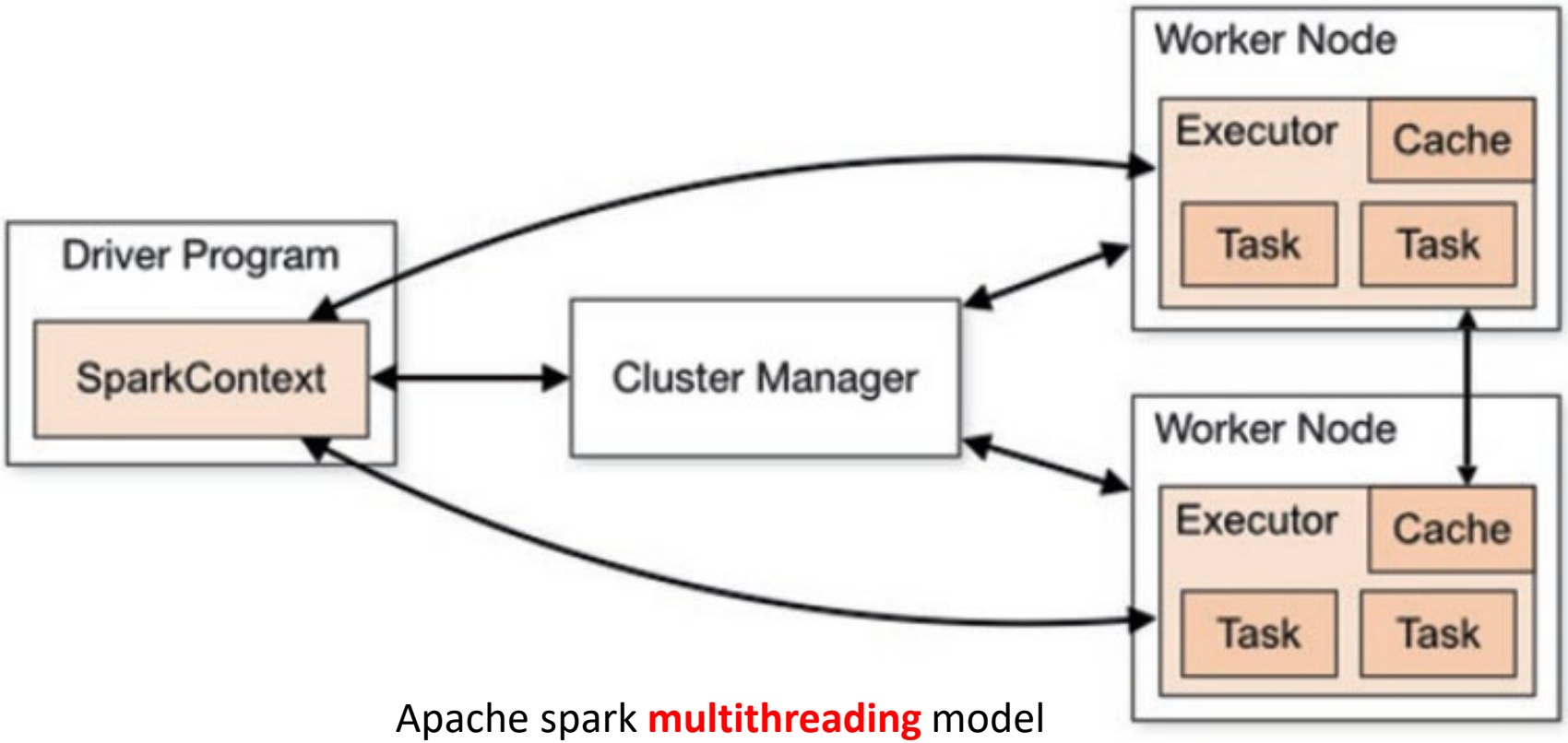
- We already discussed how Haddop/MR can have different scheduling algorithms to decide which tasks to process when there are several jobs

- Straggler issue: It can be difficult to tell the difference between a really slow and a failed replica
  - MapReduce solution:

Stream Processing

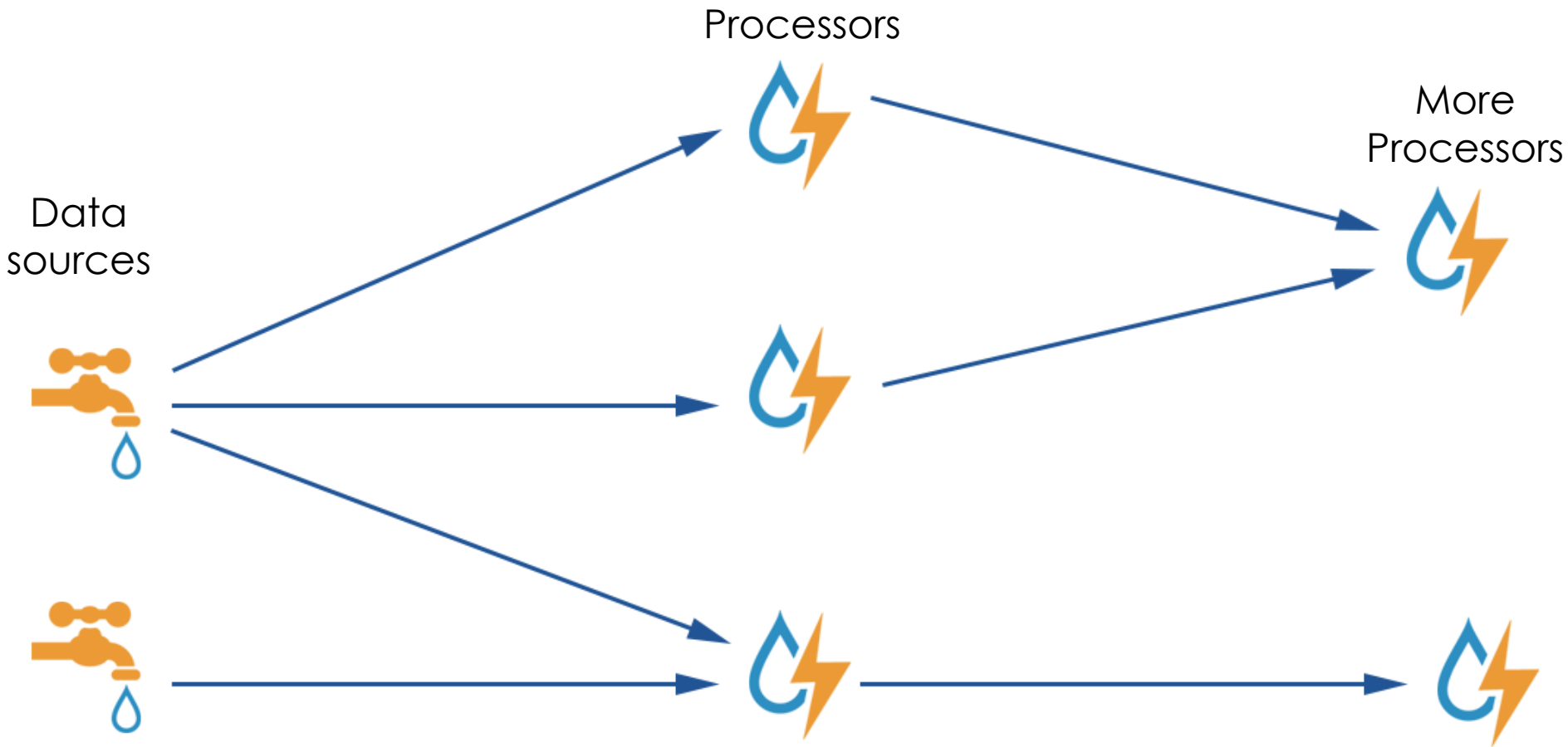
- MR/Hadoop are for **batch** processing
  - Long running jobs (minutes, hours total)
- Sometimes you want **stream** processing
  - Continuously arriving data with millisecond scale response
- **Storm** and **Spark** are basically Hadoop for streams
  - Define a graph of processing nodes
  - Stream data through the graph
  - Manage the workers (each executing a part of the graph)
  - Detect failure, carefully buffer data in queues, etc

SPARK REQUEST HANDLING MODLE



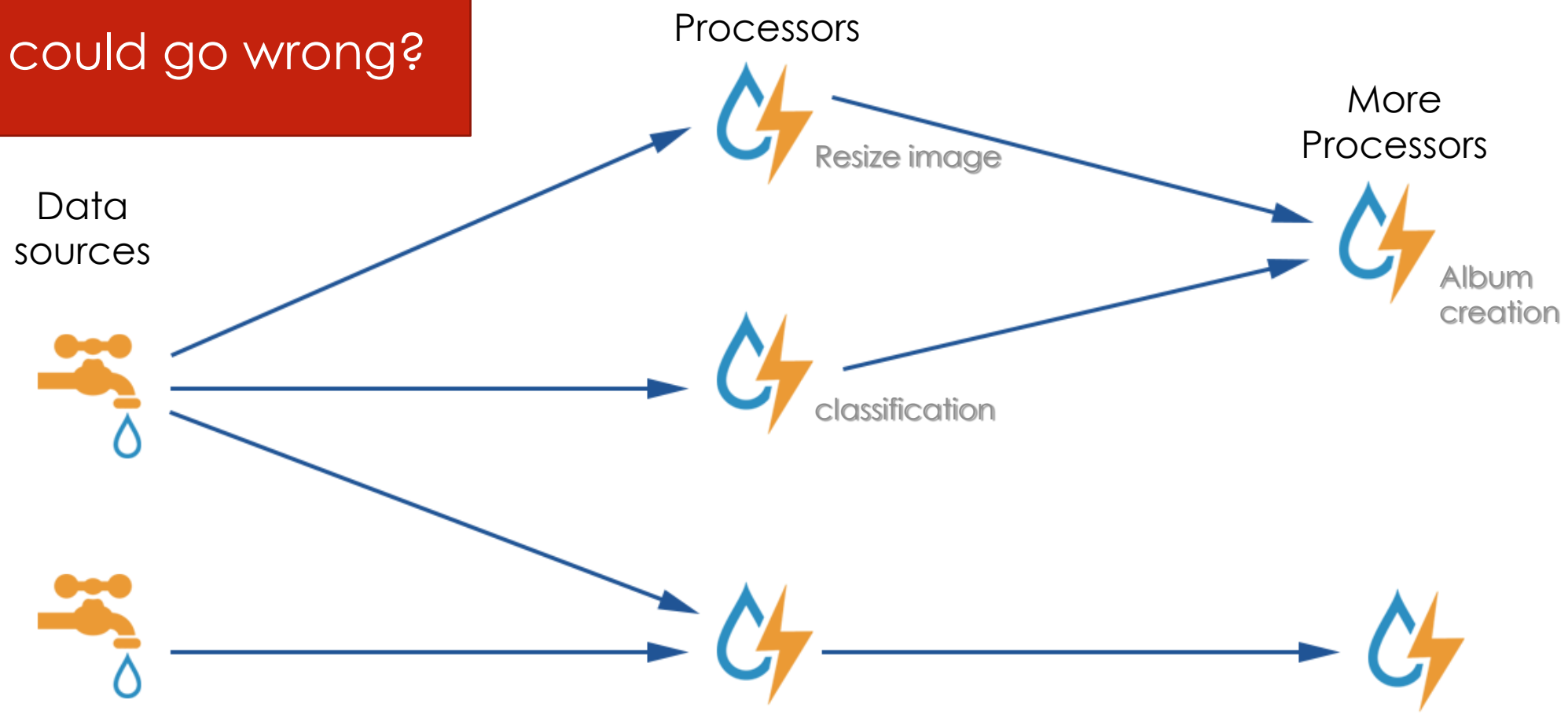
Apache spark **multithreading** model

Stream Processing



Stream Processing

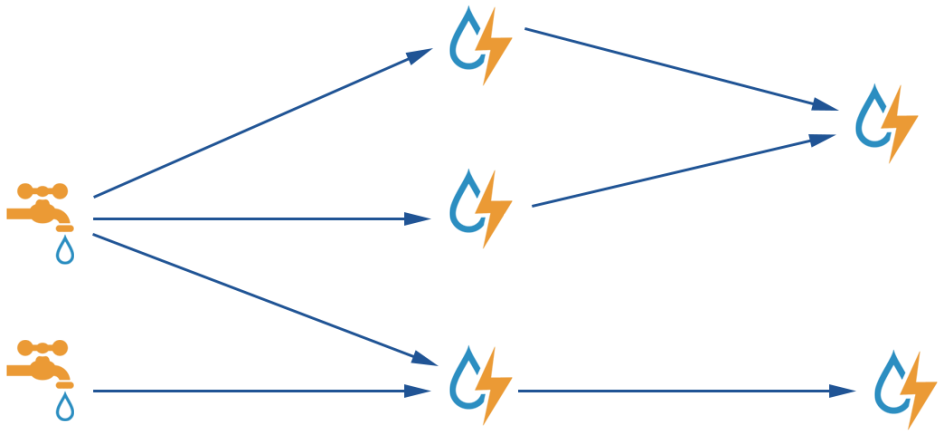
What could go wrong?



STORM **FAULT TOLERANCE**

- Data is constantly arriving -> handling faults is more difficult
- We want to understand the **fault tolerance semantics** provided by the stream processing system

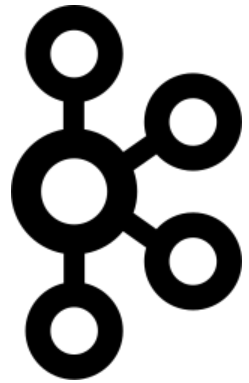
- **Best effort**: no guarantees
- **At least once**: a data element will be fully processed, but it might be processed several times
- **Exactly once**: a data element will be fully processed, precisely one time by each step



State Replication

Stream Processing **Scalability**

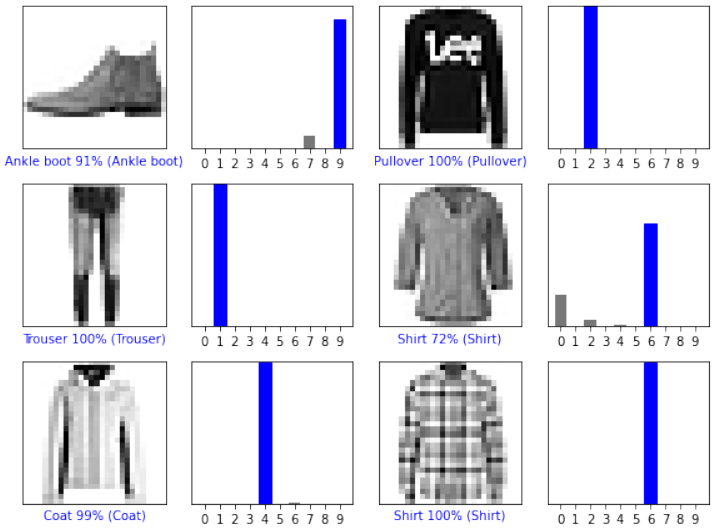
- Need to manage scalability of stream processing workers
- How to make data available to workers?
  - Hadoop Distributed File System would be too slow
- **Kafka** Message Queue
  - Distributed queue of requests
  - Queue can be scaled up and down as needed
  - Can support large numbers of workers accessing data concurrently
  - Currently uses **Zookeeper** for consistency, next version will use **Raft**



Kafka

TENSOR FLOW

- Library from Google primarily for training and running Deep Learning models
  - Designed in part by Jeff Dean (watch his talk!)
- Training machine learning models takes a lot of data
- Doing training on a single machine can be too slow
  - Or a single machine may not have enough memory
- TensorFlow handles all the ML math and helps with distributing the training and inference tasks across multiple devices



TENSOR FLOW **HETEROGENEITY**

- Many ML tasks are highly parallelizable ☺
  - But they require a lot of data -> high communication costs ☹
- GPUs are very good for this
- A network of distributed GPUs is even better…
- But custom hardware is best!

- TPU = Tensor Processing Unit
  - Customized processor specifically for tensor flow

- TensorFlow software library needs to support all of these, possibly spread across multiple servers

TENSOR PROCESSING UNIT (TPU)

- A Tensor Processing Unit (TPU) is a type of application-specific integrated circuit (ASIC) developed by Google specifically for accelerating machine learning tasks. TPUs are designed to speed up tensor operations, which are the foundational operations in deep learning computations, hence the name "Tensor Processing Unit.“
- **Characteristics of TPUs:**
  1. **High Throughput**: TPUs are designed for high volumes of low precision computation (like 8-bit integers), which is often sufficient for deep learning inference and some training tasks. This allows for a higher throughput compared to using higher precision.
  2. **Matrix Operations**: The architecture is especially well-suited for large matrix operations, which are common in deep learning. This is in contrast to traditional CPUs which handle scalar operations and GPUs that handle vector operations.
  3. **Memory**: TPUs include a large amount of high-bandwidth memory, tailored for machine learning tasks. This reduces the need to move data back and forth between the processor and memory, which can be a bottleneck.

Impact of TPU on Big Data:

1. **Accelerated Analysis**: For big data tasks that involve deep learning (like image or speech recognition in large datasets), TPUs can significantly reduce the time required to train models.
2. **Cost-Efficiency**: TPUs can lead to a reduction in the total resources (and thus costs) required for certain machine learning tasks. By speeding up computations, organizations can potentially save on infrastructure costs.
3. **Enhanced Machine Learning Models**: The performance boost provided by TPUs allows data scientists to experiment with larger, more complex models and to iterate more rapidly.
4. **Real-time Processing**: For big data applications that require real-time analysis, like autonomous vehicles or instant fraud detection, TPUs can offer the necessary computational speed.
5. **Edge Computing**: With the advent of Edge TPUs, Google has pushed for on-device machine learning, allowing for powerful ML computations on edge devices without the need to constantly communicate with a central server.
6. **Democratization of Machine Learning**: With tools like Google Colab offering free TPU usage, more people worldwide have access to high-powered machine learning resources. This has opened the door to innovation and research even for those without massive computational resources.