BIG DATA SYSTEMS

CS 564- Fall 2018

ACKs: *Magda Balazinska*

WHAT IS BIG DATA?

The three V's:

- high Volume
- high Variety
- high Velocity

VOLUME

- Databases parallelize easily
- Techniques available from the 80's (GAMMA project started @ Wisconsin!)
  - data partitioning
  - parallel query processing
- SQL is embarrassingly parallel

VARIETY

- complex workloads:
  - Machine Learning tasks: e.g. click prediction, topic modeling, SVM, k-means
- various types of data:
  - text data
  - semi-structured data
  - graph data
  - multimedia (video, photos)

VELOCITY

- data is generated very fast
- data needs to be processed very fast

- real time data analytics
- data streaming (each data item can be processed only once!): e.g. financial data

ANOTHER V: VERACITY

The data collected is often uncertain

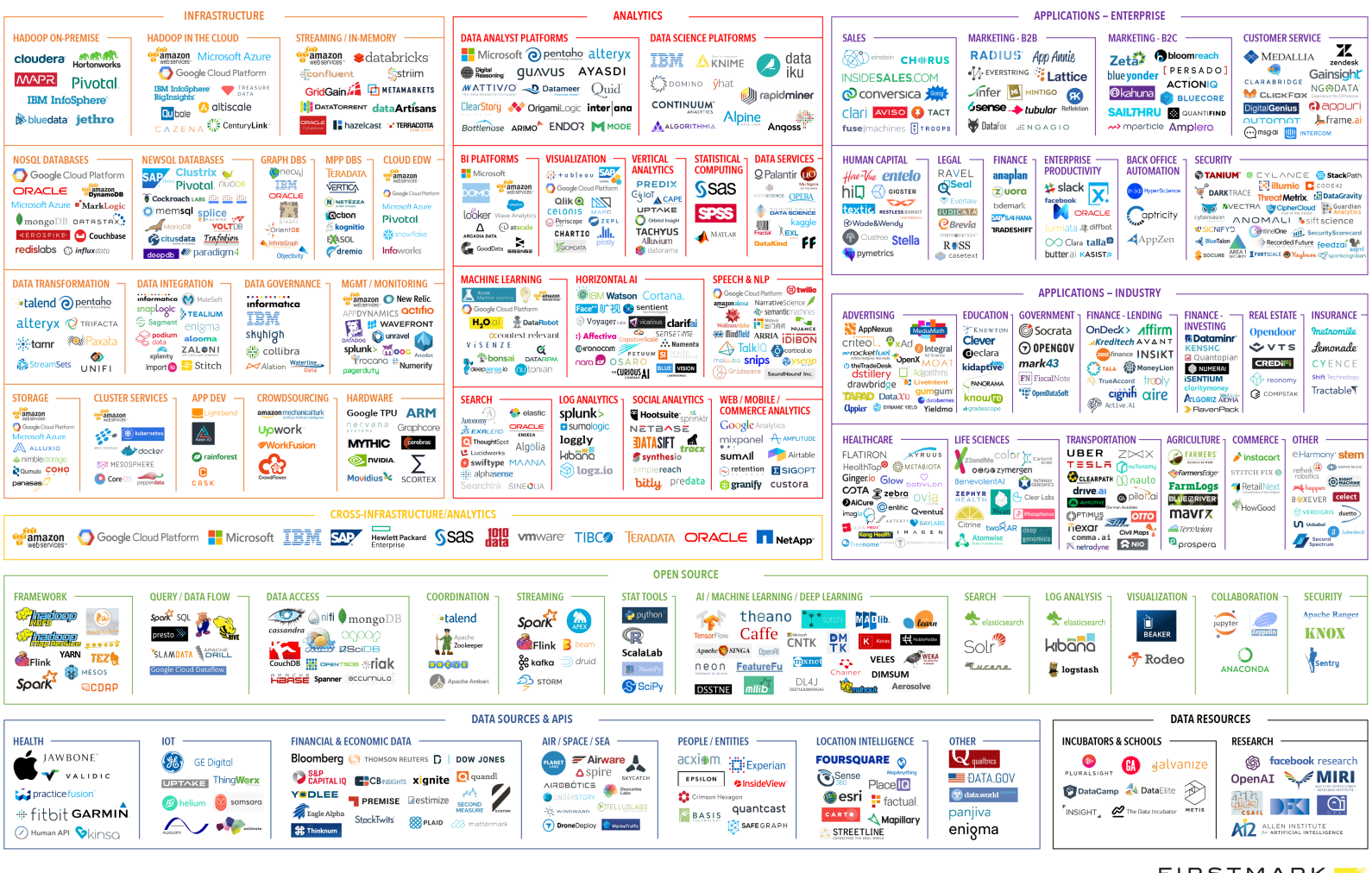
- inconsistent data (violated constraints)
- incomplete data (missing values)
- ambiguous data

Example: sensor data

DATA ANALYTICS COMPANIES

- Greenplum: founded in 2003 acquired by EMC in 2010. A parallel shared-nothing DBMS
- Vertica: founded in 2005 and acquired by HP in 2011. A parallel column-store shared-nothing DBMS
- AsterData: founded in 2005 acquired by Teradata in 2011. A parallel, shared-nothing, MapReduce-based data processing system
- Netezza: founded in 2000 and acquired by IBM in 2010. A parallel shared-nothing DBMS

THE BIG DATA LANDSCAPE



SCALING FOR BIG DATA

- **OLTP**: Online Transaction Processing
  - scale transactions per second
  - Amazon, Facebook, Twitter, …
- **OLAP**: Online Analytical Processing
  - scale query response time
  - analysis of massive datasets, decision making, …

2 APPROACHES

- Parallel databases, started at the 80s. Both for
  - OLTP (transaction processing)
  - OLAP (decision support queries)
- MapReduce
  - first developed by Google, published in 2004
  - only for decision support queries
  - ecosystem around it: Hadoop, PigLatin, Hive, …

Today these two approaches have been converging!

PARALLEL DBMS

- The goal is to improve performance by executing multiple operations in parallel

- Terminology to measure performance:
  - **Speed-up**: using more processors, how much faster does the task run (if problem size is fixed)?
  - **Scale-up**: using more processors, does performance remain the same as we increase the problem size?

SCALE-UP VS SCALE-OUT

Scale-up

- using more powerful machines, more processors/RAM per machine

Scale-out

- using a larger number of servers

ARCHITECTURES

- Shared memory
  - nodes share RAM + disk
  - easy to program, expensive to scale
- Shared disk
  - nodes access the same disk, hard to scale
- Shared nothing
  - nodes have their own RAM+disk
  - connected through a fast network

PARALLEL QUERY EVALUATION

- Multiple DBMS instances also called nodes execute on machines in a cluster
  - one instance plays role of the coordinator
  - other instances play role of workers
- Workers execute queries
  - typically all workers execute the same plan (intra-operator parallelism, intra-query parallelism)
  - workers can execute multiple queries at the same time (inter-query parallelism)

PARALLEL DATA STORAGE

Horizontal data partitioning

- block partitioned
- hash partitioned
- range partitioned

Uniform vs skewed partitioning

PARALLEL QUERY ALGORITHMS

- Parallel Selection

- Parallel Join
  - hash join
  - broadcast join

BEYOND PARALLEL DBMS

Many big data systems go beyond the capabilities of relational systems:

- MapReduce/Hadoop
- Spark, SparkSQL
- Mlib, Tensorflow
- Graph processing systems
- Stream processing systems