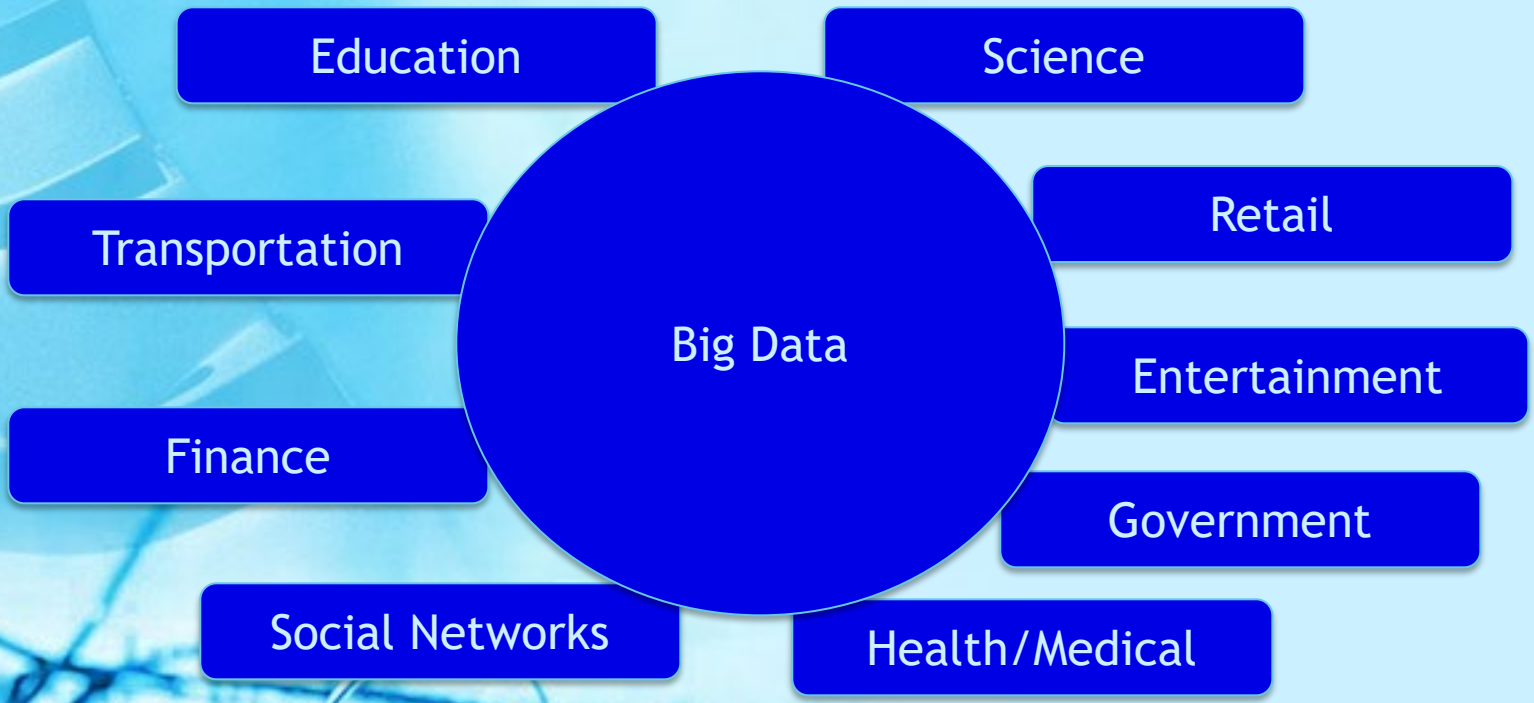
Big Data
and
No-SQL/New SQL Databases

April 29, 2024

What is Big Data?

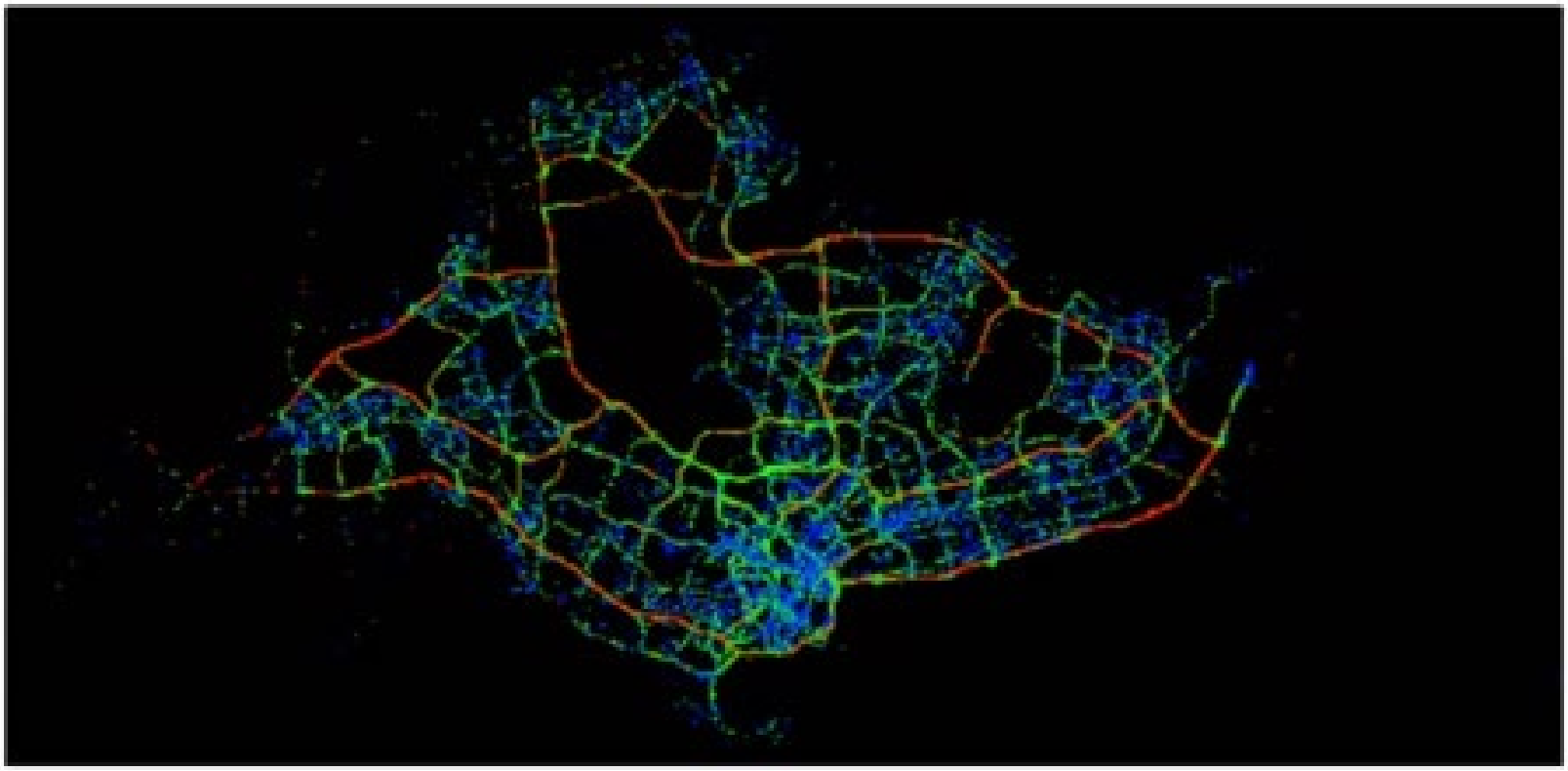
- Extremely large data sets
- Hard to manage volumes of data
- Structured, semi-structured, unstructured data
- Computationally analyze to find patterns, trends and associations

Sources of Big Data



Application: Congestion-aware Routing

Traffic Visualization – Speed/Congestion

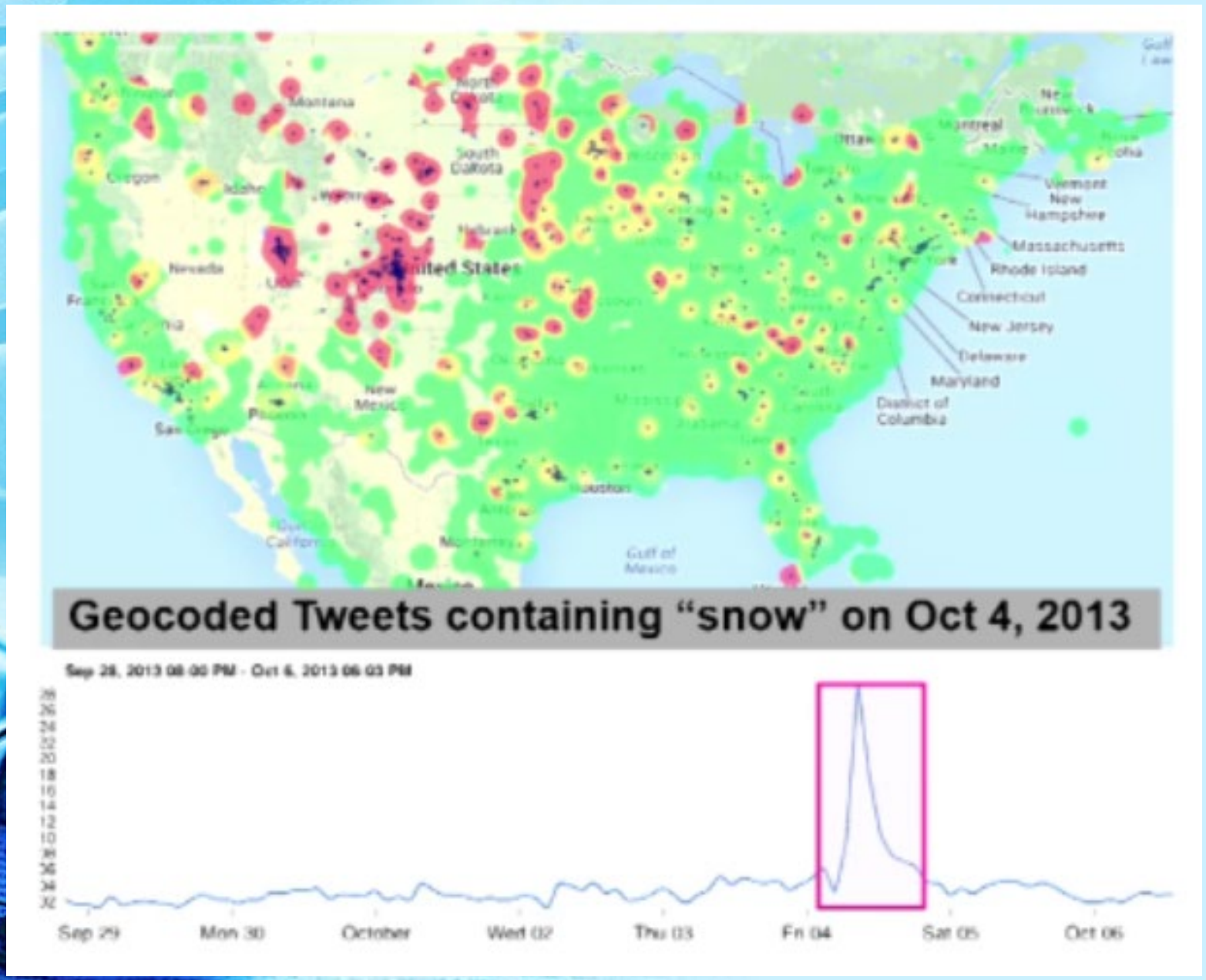


Slow Medium Fast

Twitter stats

- Number of people who use Twitter: 450 million (as of 2022)
- 6000 tweets per second → 500 million tweets each day → 200 billion each year
- More than just 280 characters:
  - Geo coordinates
  - Timestamp
  - User and follower information
  - Reply information
  - Hashtags
  - Device/platform used to post

Geocoded data



Transactions vs. Analytics

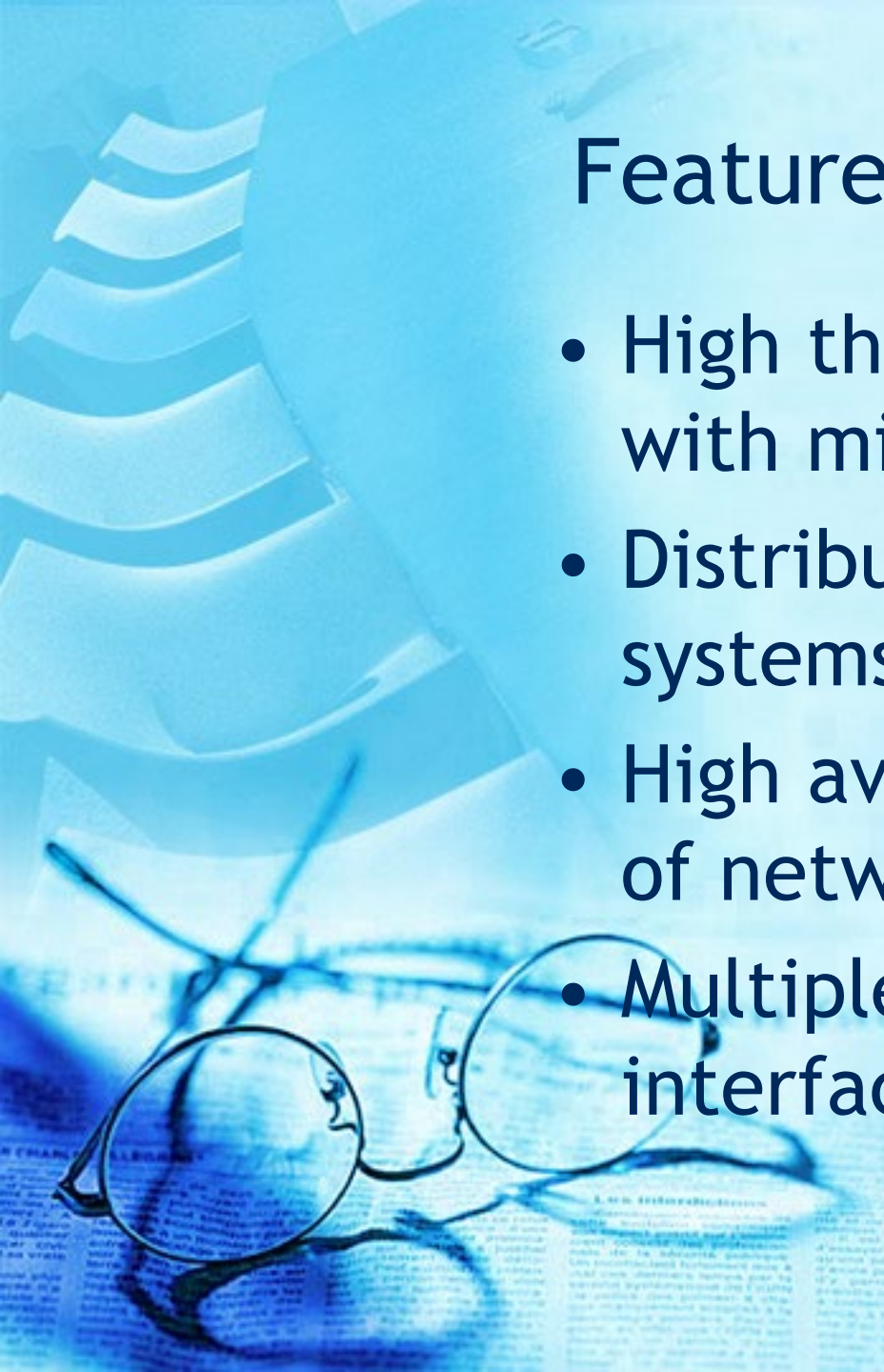
- Transactions
  - Lots of small reads, writes of individual records
  - Several concurrent users
  - Non-sophisticated processing
- Analytics
  - Analysis of large fractions of a database
  - Mostly reads
  - Complex processing

Features in a transactional database

- Record-oriented persistent storage
- Carefully structured data
- Powerful query language
- Transactions
  - Groups of statements executed together

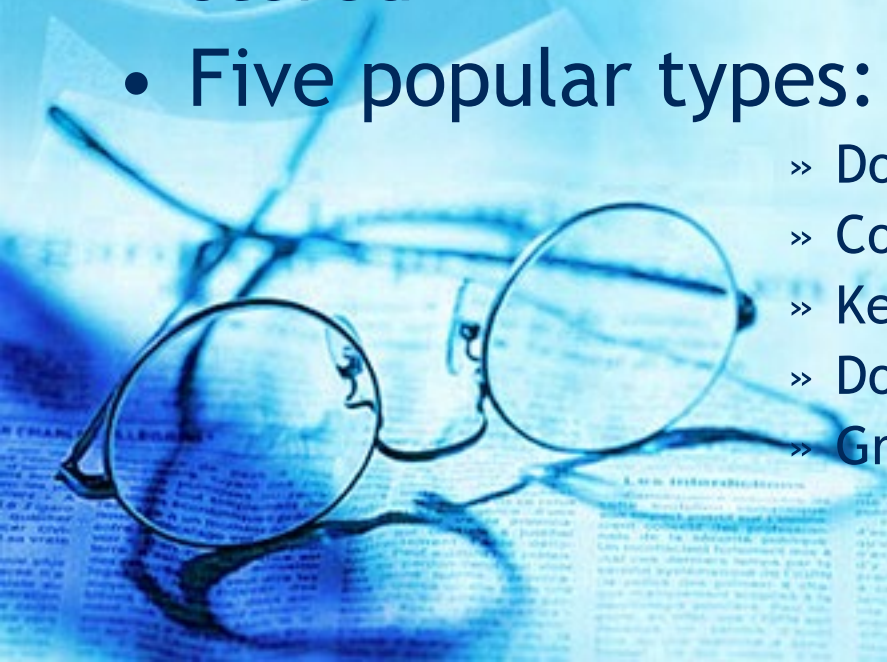
Features needed

- High throughput operation with millions of users
- Distributed across many systems
- High availability, irrespective of network failures
- Multiple programming interfaces/data models



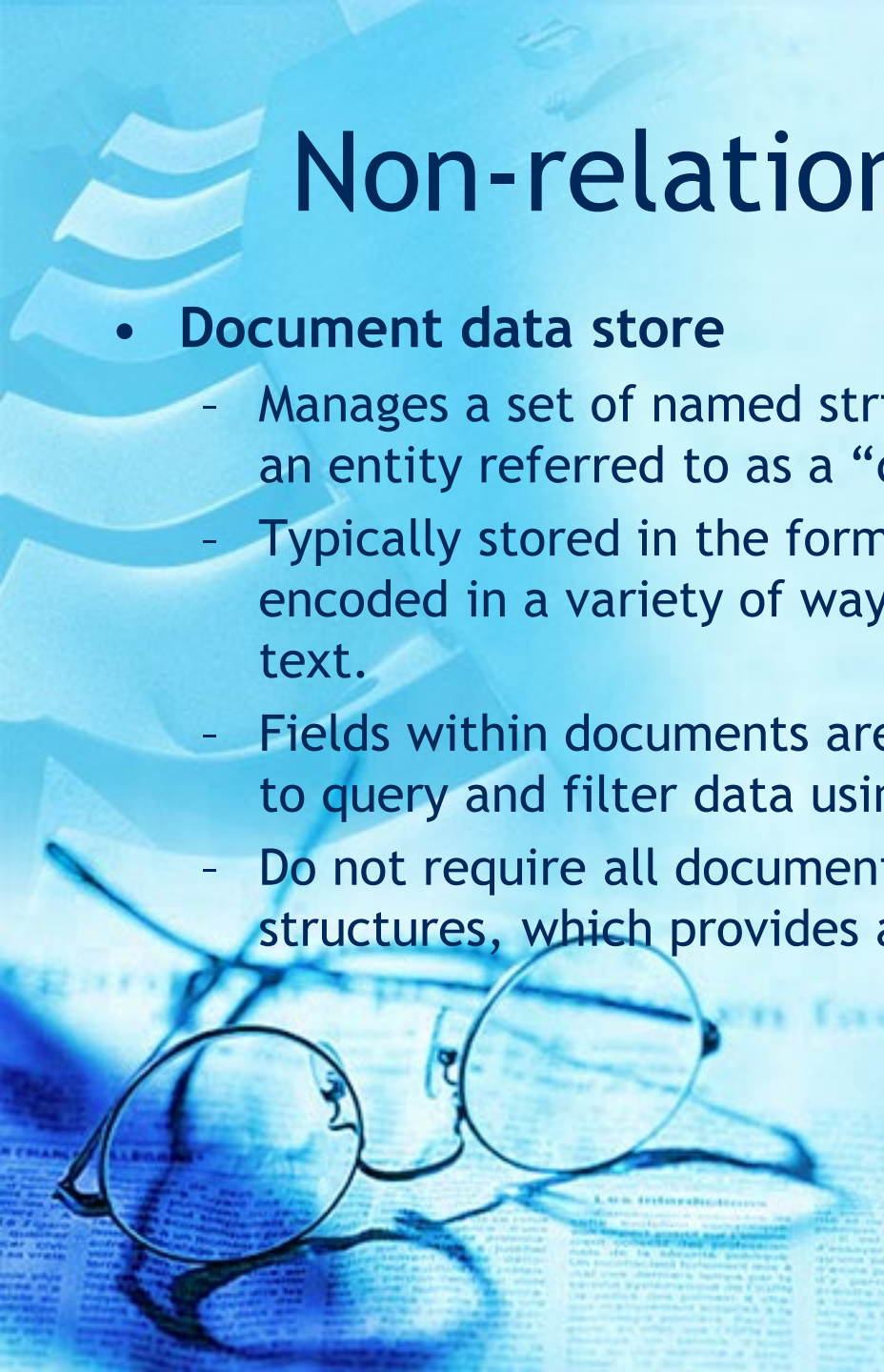
Non-relational databases

- Store data – without using tables, rows, primary keys or foreign keys
- Storage model optimized for specific requirements of the type of data to be stored
- Five popular types:
  - Document data store
  - Column-oriented database
  - Key-value store
  - Document store
  - Graph database



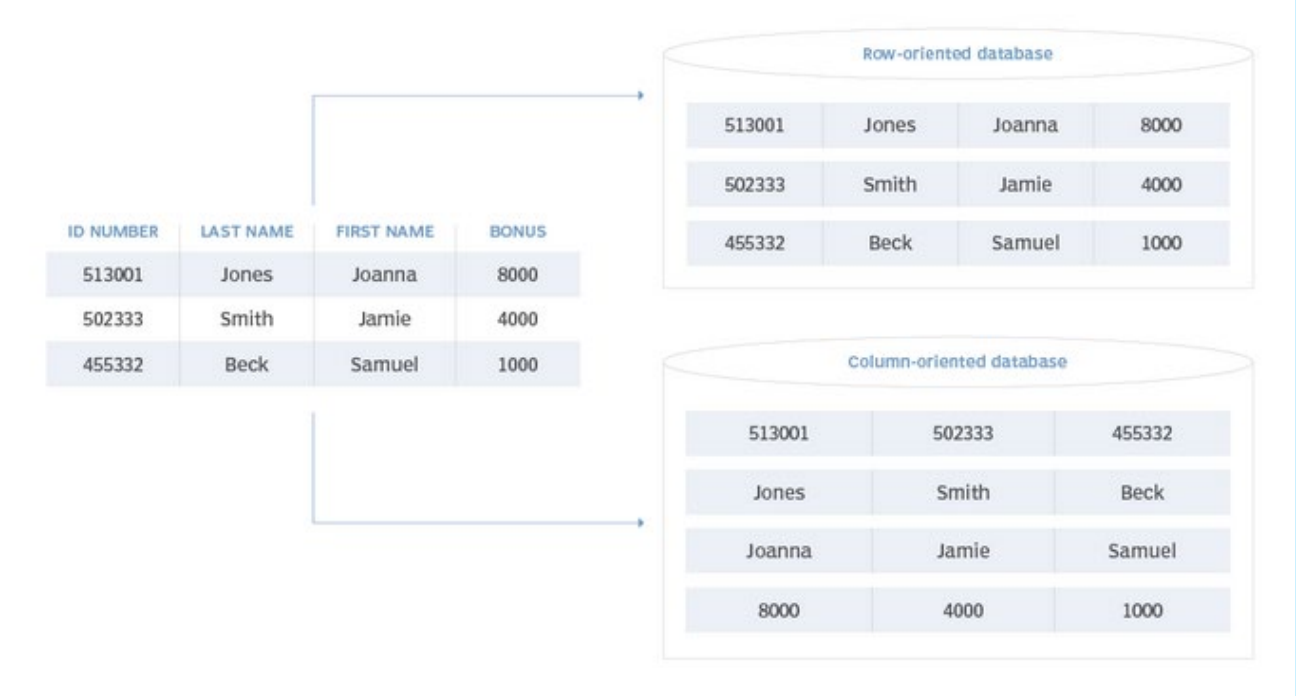
Non-relational databases

- **Document data store**
  - Manages a set of named string fields and object data values in an entity referred to as a “document”
  - Typically stored in the form of JSON documents, which can be encoded in a variety of ways, including XML, JSON or as plain text.
  - Fields within documents are exposed, allowing an application to query and filter data using field values.
  - Do not require all documents to maintain identical data structures, which provides a great deal of flexibility.



Non-relational databases

- **Columnar (column-oriented) data store**
  - Organize data into columns, conceptually similar to the relational model
  - Denormalized approach to structuring sparse data
    - comes from its column-oriented approach to storing data



Non-relational databases

- **Key Value Store**
  - Least complicated model
  - Data mapped from keys to arbitrary values
  - No conformity to any particular structure
  - Eg. MongoDB, Apache Cassandra, Redis, Couchbase and Apache Hbase
  - Limited multi-record transactional consistency

Key	Value
6.01	"Intro to EECS 1 by Professor Madden"
6.02	"Intro to EECS 2 by Professor Stonebraker"
6.033	"Systems by Professor Zeldovich"
6.814	"Databases by Professor Smith"

get("6.01") → "Intro to EECS 1 by Professor Madden"
put("6.005","Software Engineering by Prof. Jones")

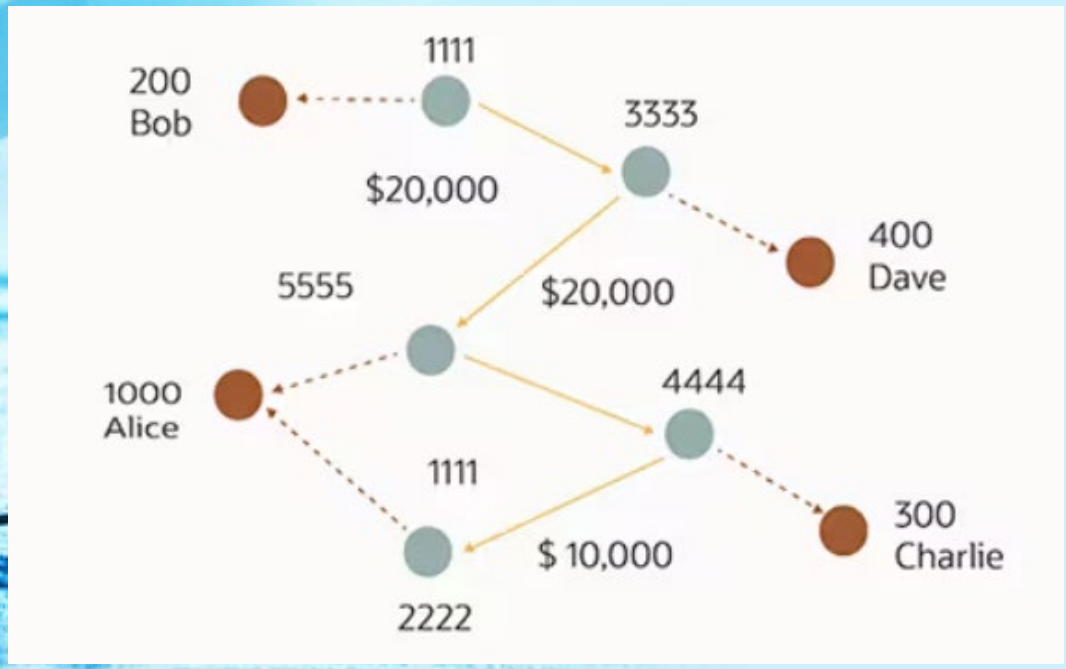
Non-relational databases

- **Document Store**
  - More complex than key value store
  - Data are a mapping of keys to (XML/JSON) documents
  - Store everyday documents as is
  - Eg. MongoDB, CouchDB
  - Can lookup documents by key
  - Can search contents of documents
  - Joins or multi-document updates ??

Key	Value
6.01	{title: "Intro To EECS", prof: "Madden", room: 123}
6.02	{title: "Intro To EECS 2", professor: "Stonebraker"}
6.033	{title: "Systems", room: 145}
6.814	{title: "Databases, room: 154, professor: "Smith"}

Non-relational databases

- **Graph database**
  - Most complex of the non-relational databases
  - Efficiently store relations between entities, facilitating greatly interconnected data
  - Eg. Purchasing/manufacturing systems

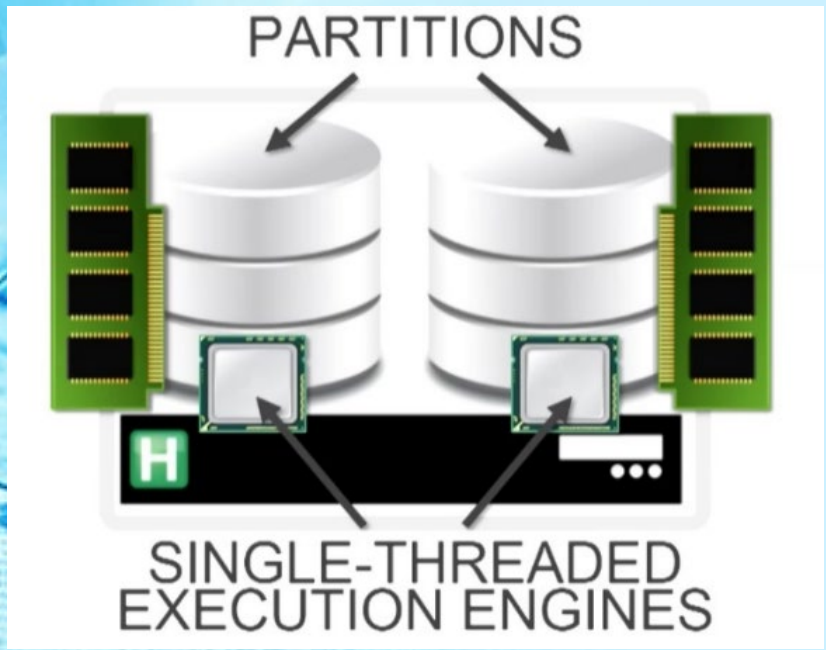


New SQL Databases

- Partition DB into RAM-sized chunks
- Store in memory of a cluster of machines
- OLTP workloads partition well
  - Per customer shopping carts
  - Per user email accounts
- Partitioning → Parallelism

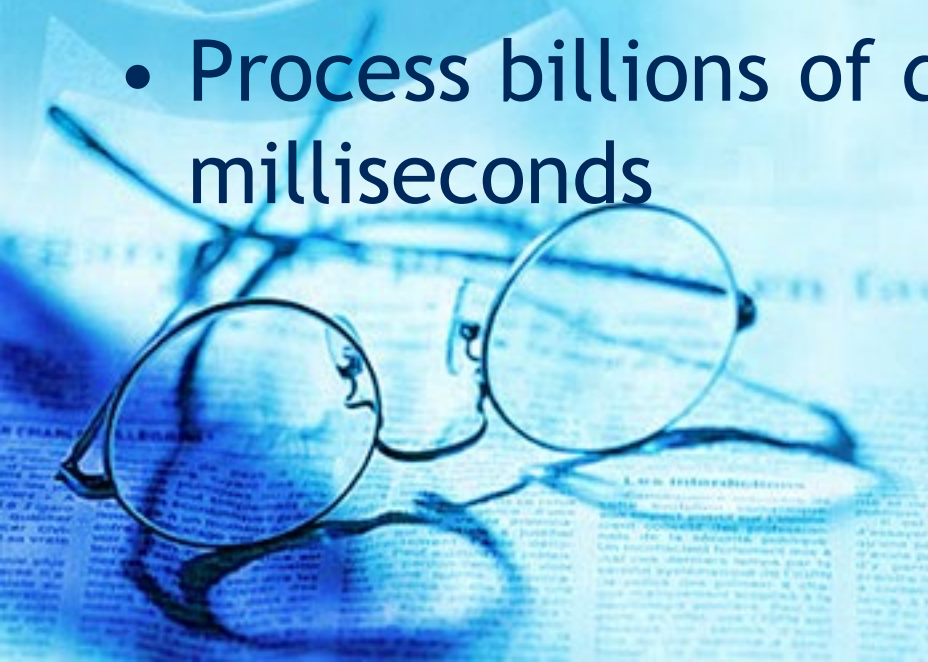
New SQL Databases

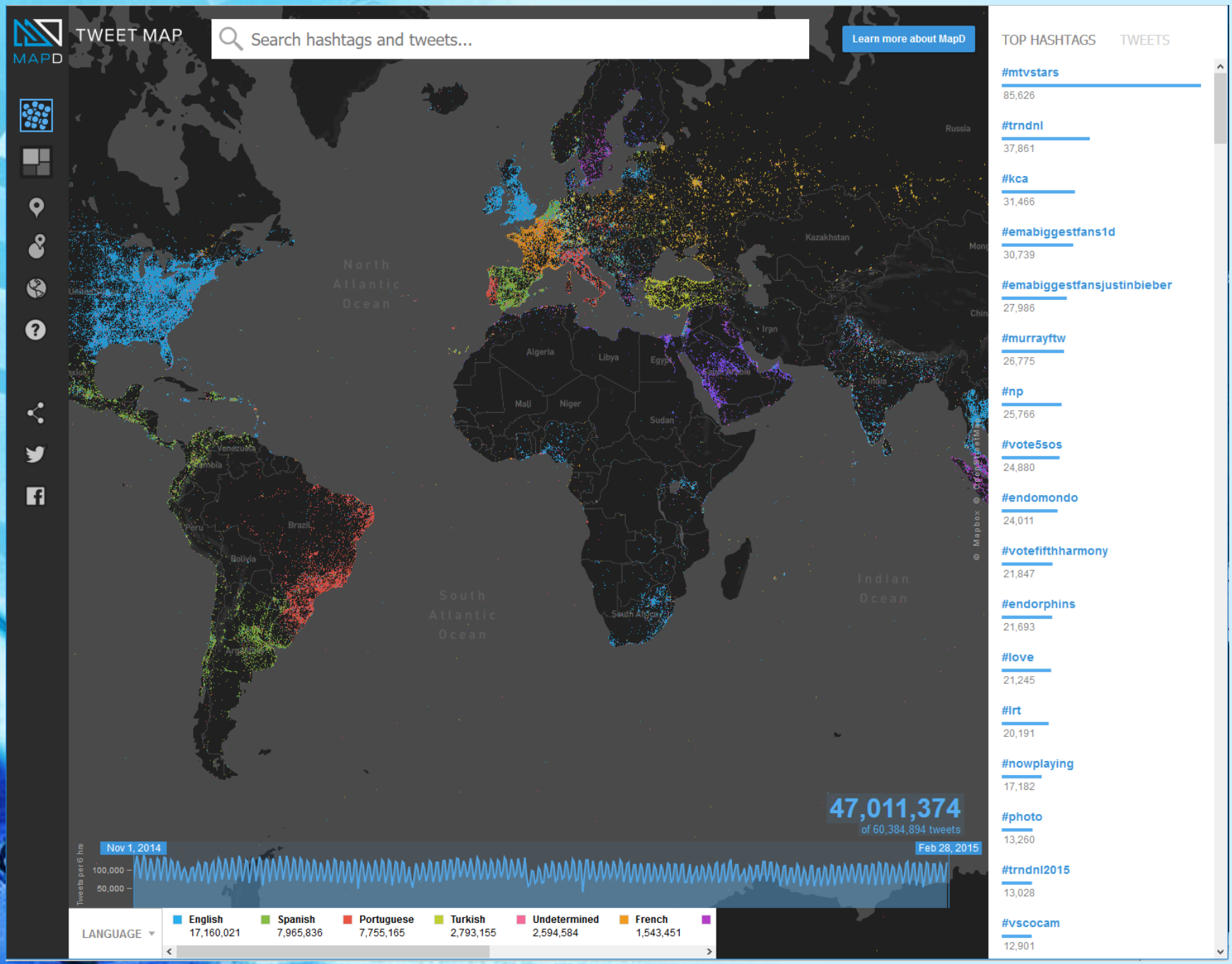
- Main memory storage
- Serial execution
- Compact logging



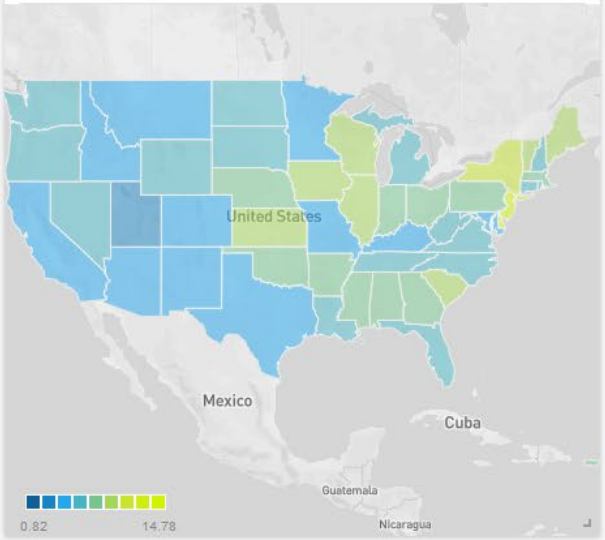
MapD – Massively Parallel Database

- To make big data exploration interactive and insightful
- Leverage the massive bandwith and parallelism of GPUs
- Process billions of data points in milliseconds



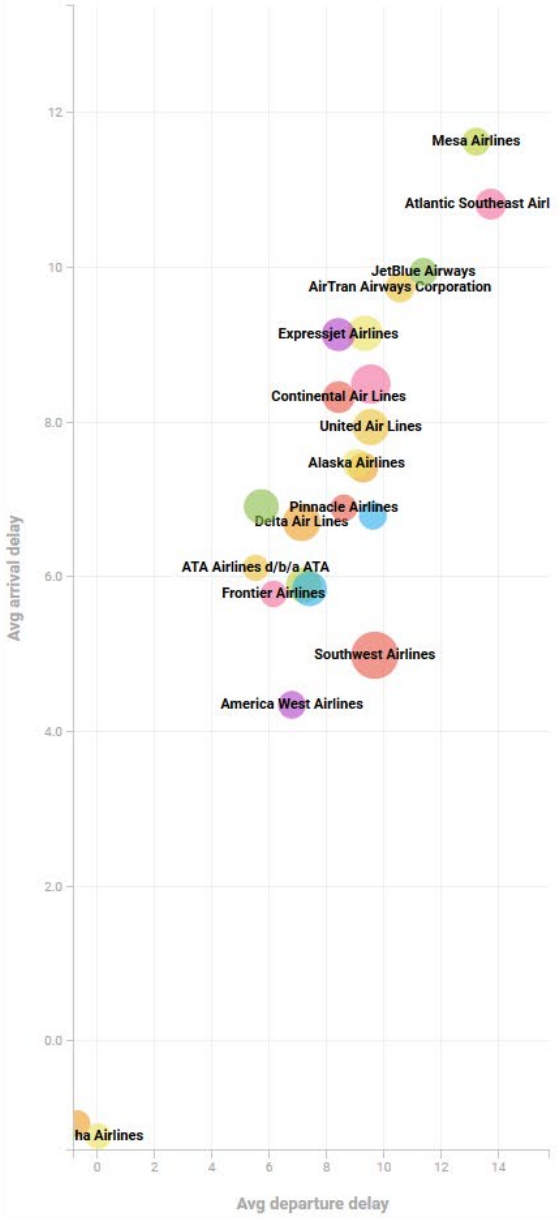

TWEET MAP
MAPD
Search hashtags and tweets...
Learn more about MapD
TOP HASHTAGS
TWEETS
#mtvstars
85,626
#trndnl
37,861
#kca
31,466
#emabiggestfans1d
30,739
#emabiggestfansjustinbieber
27,986
#murrayftw
26,775
#np
25,766
#vote5sos
24,880
#endomondo
24,011
#votefifthharmony
21,847
#endorphins
21,693
#love
21,245
#irt
20,191
#nowplaying
17,182
#photo
13,260
#trndnl2015
13,028
#vscocam
12,901
Russia
Kazakhstan
Iran
North Atlantic Ocean
South Atlantic Ocean
Indian Ocean
Algeria
Libya
Egypt
Mali
Niger
Sudan
Brazil
Bolivia
Peru
Venezuela
47,011,374
of 60,384,894 tweets
Nov 1, 2014
Feb 28, 2015
Tweets per 6 hrs
100,000
50,000
LANGUAGE
English 17,160,021
Spanish 7,965,836
Portuguese 7,755,165
Turkish 2,793,155
Undetermined 2,594,584
French 1,543,451

Average Arrival Delay By Destination State

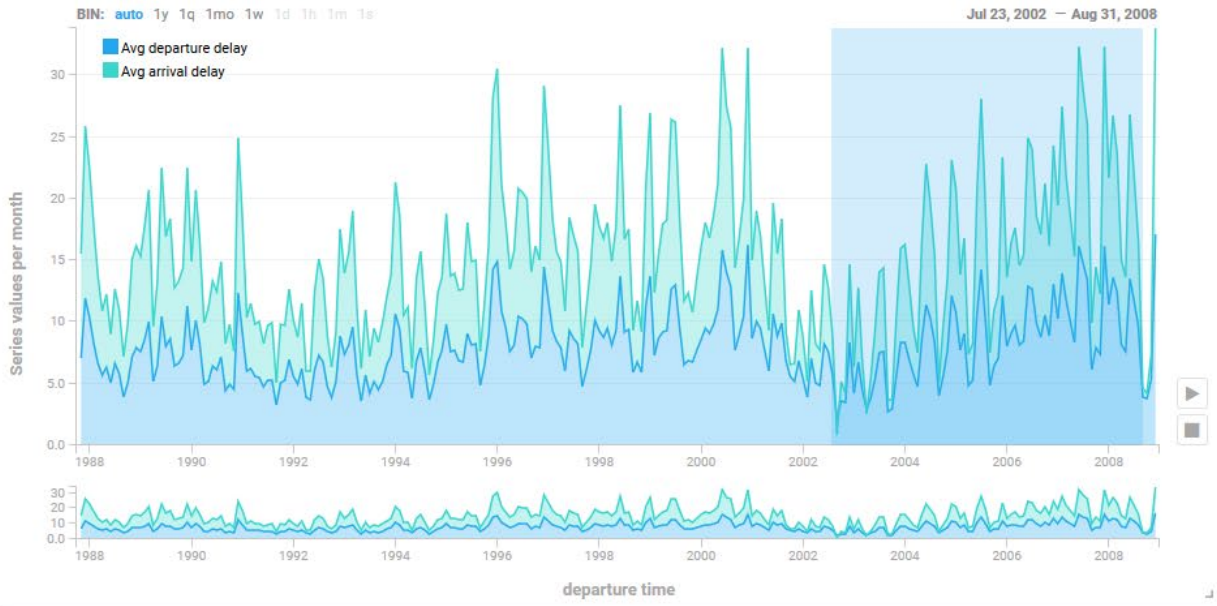


Count By Month, Day Of Week

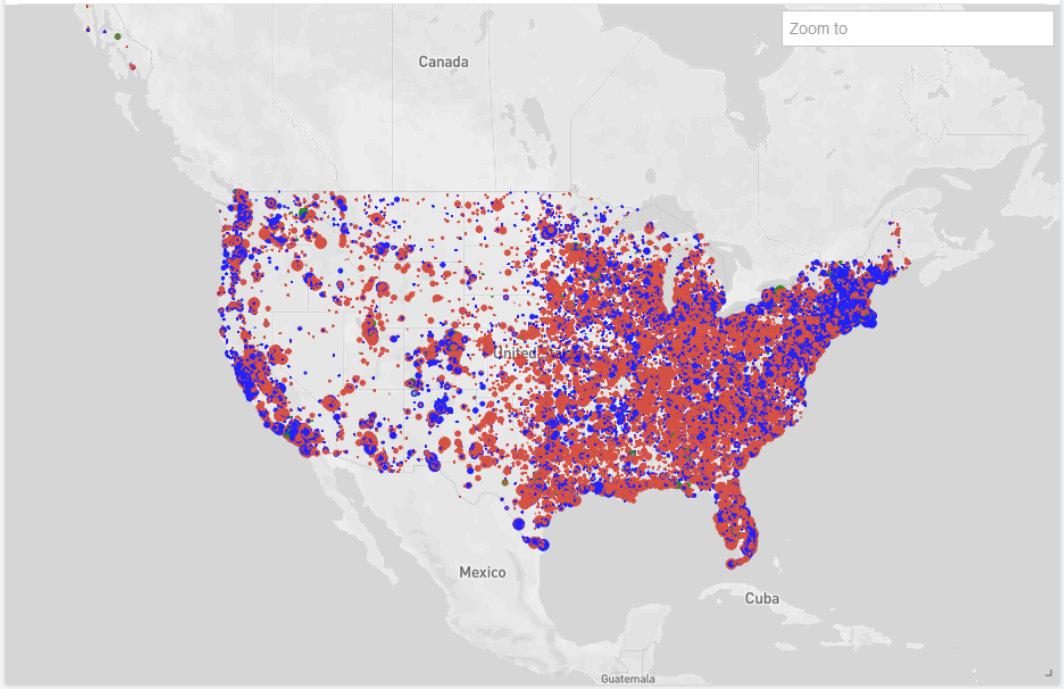
Average Departure Delay Vs. Average Arrival Delay By Airline Name



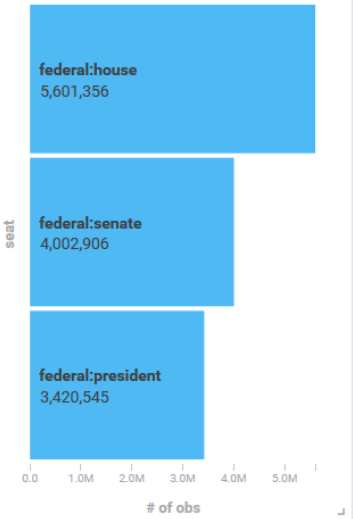
Average Departure Delay Vs. Average Arrival Delay By Departure Time (Month)



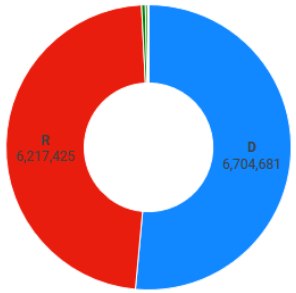
Point Map



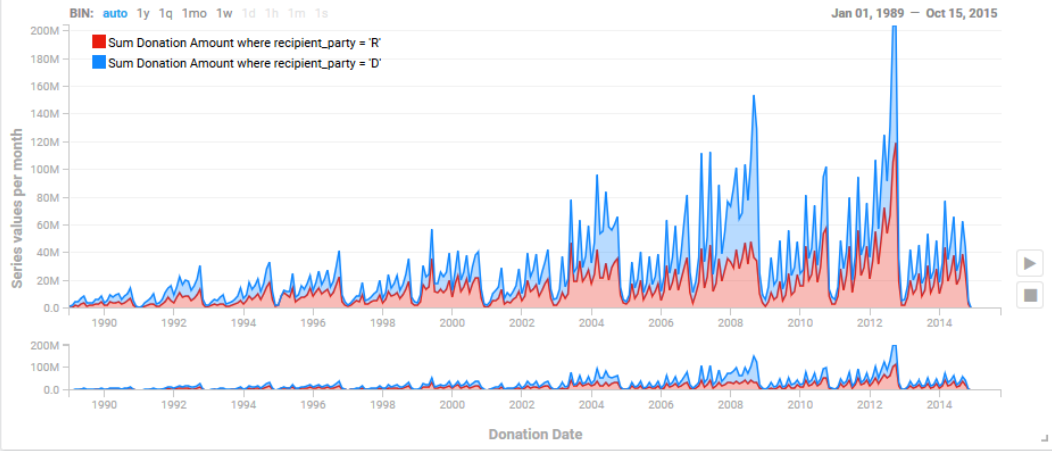
Count By Seat



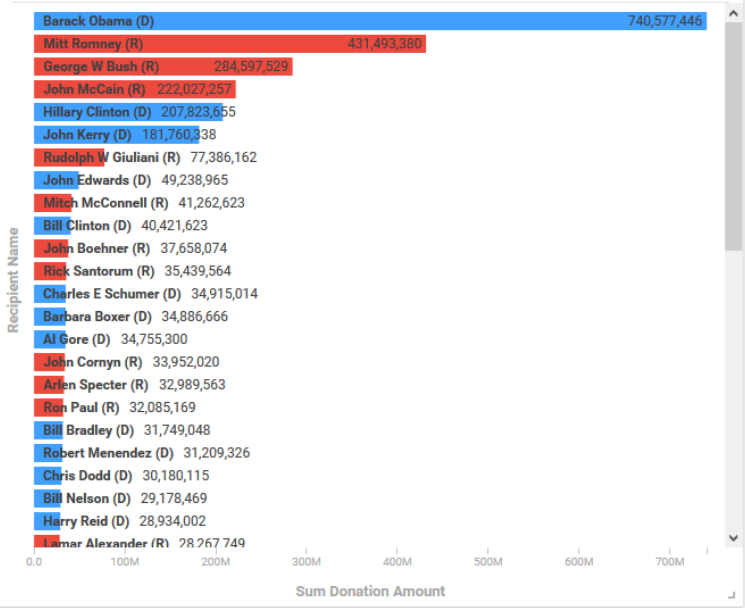
Count By Recipient Party



Sum Of Donation Amount Where Recipient_party = 'R' Vs. Sum Of Donation Amount Where Recipient_party = 'D' By Donation Date (Month)



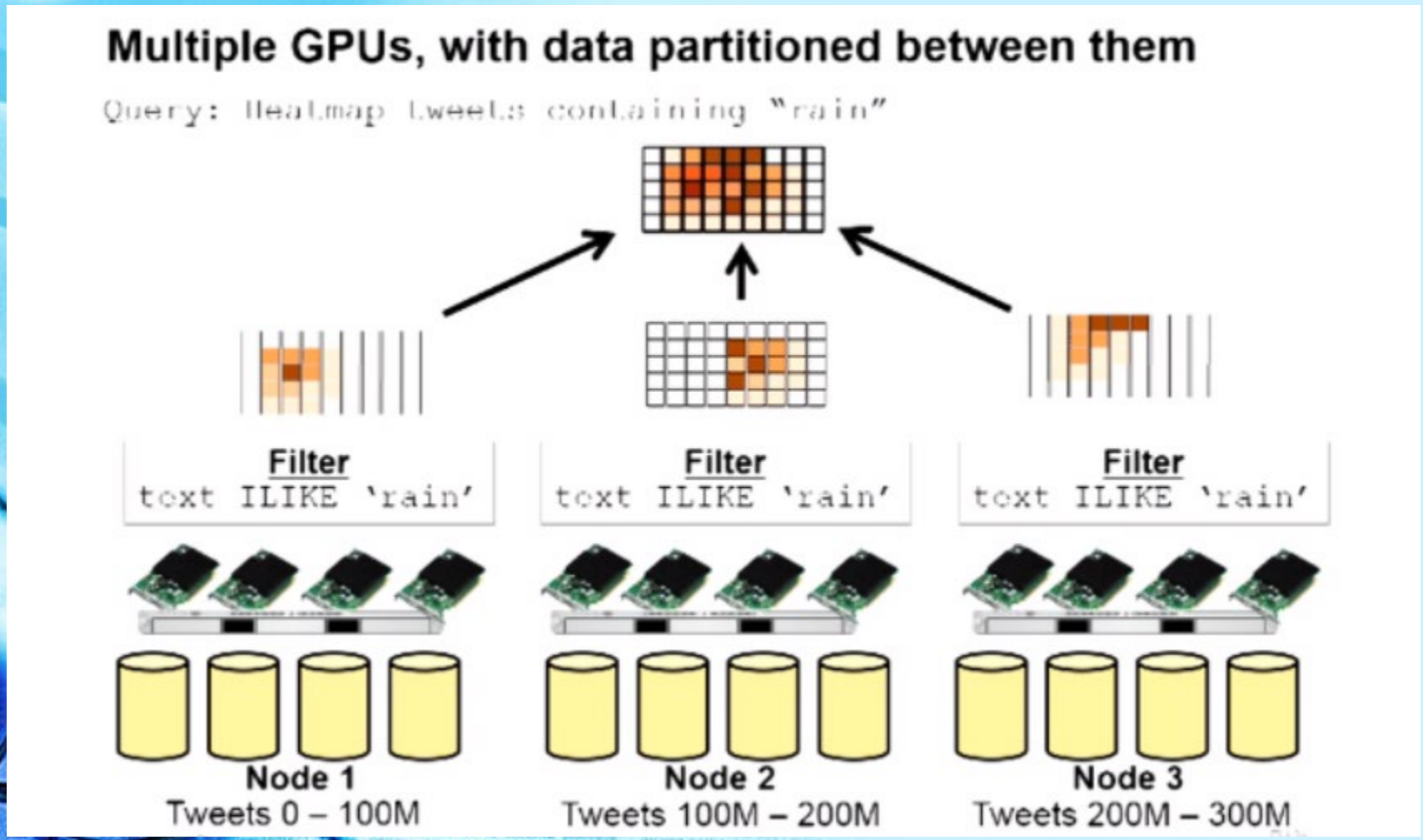
Sum Of Donation Amount By Recipient Name



How does MapD work?

- GPUs have enough memory that a cluster of them can store large amounts of data
- Not an accelerator, but a full-blown SQL database!
- Effective parallel processing

“Shared Nothing” Processing



Relational Databases

- They work with structured data; table and row-oriented
- Relationships in the system have constraints, which promotes a high level of data integrity
- Support indexing capabilities, resulting in faster query response times
- Excellent at keeping data transactions secure
- Facilitate complex SQL queries for data analysis and reporting
- These models ensure and enforce business rules at the data layer adding a level of data integrity not found in a non-relational database
- Use SQL (structured query language) for shaping and manipulating data
- SQL databases are best fit for heavy duty transactional type applications

Non-Relational Databases

- Document oriented
- Ability to store large amounts of data with little structure or no structure limitations
- Provide scalability and flexibility for changing business needs
- Provide schema-free or schema-on-read options
- Store all types of data ‘Big Data’ including unstructured data
- Flexible data model with the ability to easily store and combine data of any structure without the need to modify a schema

COSC-3380

Final Exam Review

Date/Time/Location

Location: SEC 103

Date: Monday, May 6, 2024

Time: 6 PM to 8 PM

Final Exam Review

Complex SQL

- Nested Queries
- Aggregate functions
- Assertions
- Triggers
- Views
- Joins

Entity Relationship (ER) model

Testing your ability to read an ER diagram!

Normalization

Informal design guidelines

Functional Dependencies

Normal Forms

- 1NF
- 2NF
- 3NF
- BCNF

Database Security

All aspects of security

- Threats
- Steps to prevent and protect data
- SQL injection

Relational vs. Non-relational

Differences

When to use what type of DB?