THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

12. NoSQL

CSCI 2541 Database Systems & Team Projects

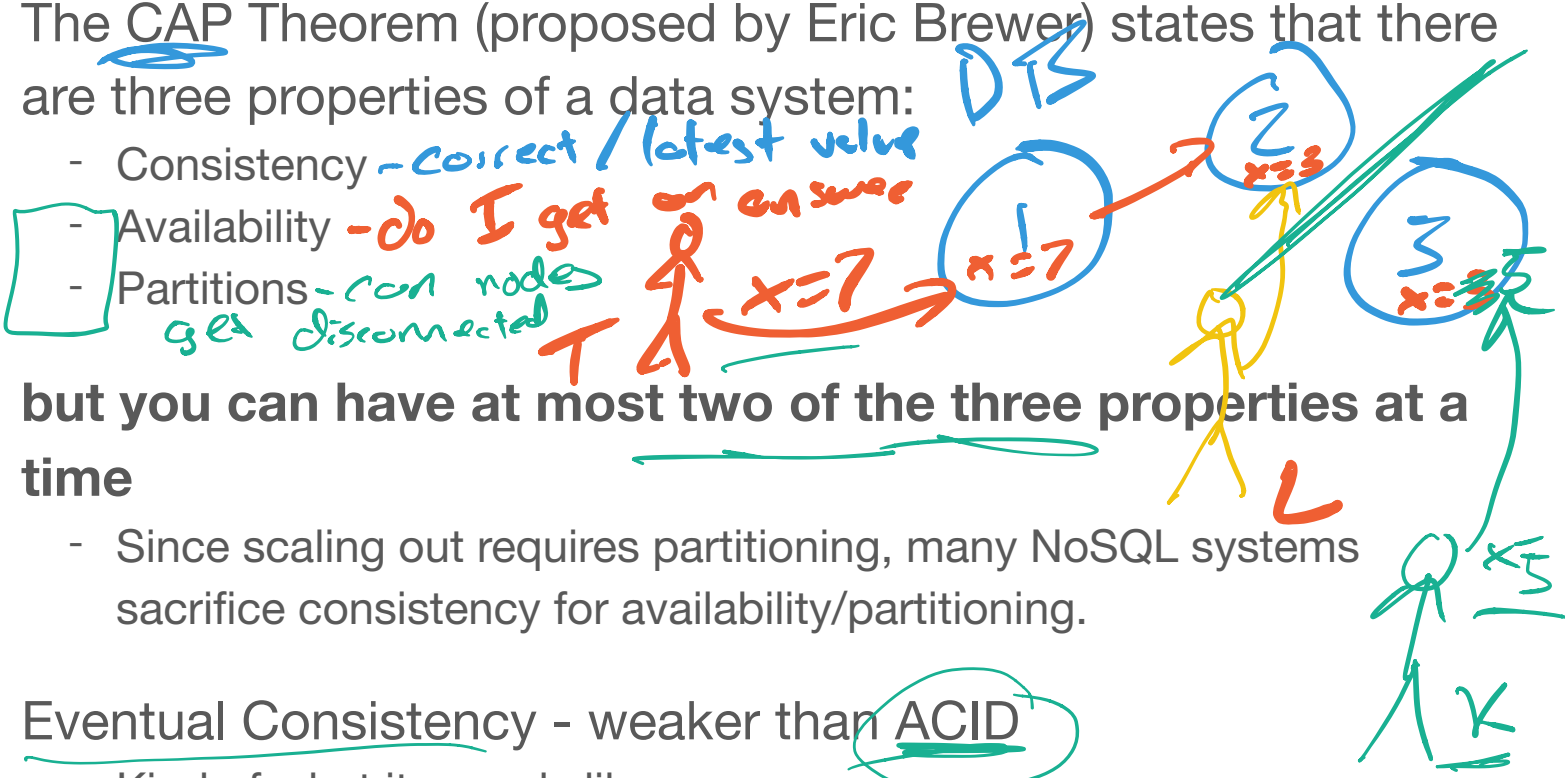
Wood & Chaufournier

Includes slides from Prof. Bhagi Narahari

CAP Theorem..getting around ACID

The CAP Theorem (proposed by Eric Brewer) states that there are three properties of a data system:

- Consistency
- Availability
- Partitions



but you can have at most two of the three properties at a time

- Since scaling out requires partitioning, many NoSQL systems sacrifice consistency for availability/partitioning.

Eventual Consistency - weaker than ACID

- Kind of what it sounds like
- Does not guarantee updates are immediately visible
- But eventually all nodes will agree on a final value

NoSQL (Data) Models

There are a variety of models/systems that are not relational:

- Column Stores – represent data in columns rather than rows.
  - Examples: Google BigTable, HBase, Cassandra
- Key-value stores – ideal for retrieving specific data records from a large set of data
- Document stores – similar to key-value stores except value is a document in some form (e.g. JSON)
- Graph databases – represent data as graphs

Related:

- MapReduce – technique for large scale data analysis provided by many NoSQL DBMSs

Typical NoSQL API

Basic API access:

- **get**(key) -- Extract the value given a key
- **put**(key, value) -- Create or update the value given its key
- **delete**(key) -- Remove the key and its associated value
- **execute**(key, operation, parameters) -- Invoke an operation to the value (given its key) which is a special data structure (e.g. List, Set, Map .... etc).

What is missing compared to SQL?

- ✗ WHERE
- ✗ Grouping
- ✗ Fine grain attributes
- ✗ JOIN

What do you lose with NoSQL systems?

Joins, group by; order by

- Implement this logic in the application layer (eg Python)

ACID transactions

SQL

Enterprise integration with other relational and SQL-based systems

JDBC/ODBC APIs

familiarity and standards compliance

1. Key-Value Data Model

Key-value stores store and retrieve data using keys. The data values are arbitrary. Designed for "web sized" data sets.

Operations:

- insert(key, value), fetch(key), update(key), delete(key)

Basically just a remote Dictionary / Hash Table / Hashmap

Benefits: high-scalability, availability, and performance

Limitations: single record transactions, eventual consistency, simple query interface

Examples: Cassandra, Amazon Dynamo, Google BigTable, HBase, Redis, memcached

2. Document Store Data model

Document stores are similar to key-value stores but the value stored as a structured document (e.g. JSON, XML).

Can store and query documents by key as well as retrieve and filter documents by their properties

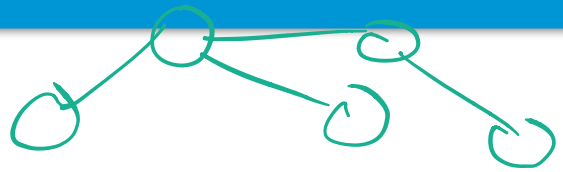
- Not as powerful as SQL

Benefits: high-scalability, availability, and performance

Limitations: same as key-value stores, may cause redundancy and more code to manipulate documents

Examples: CouchDB, SimpleDB, MongoDB, Document DB

3. Graph Databases



Model the data as a graph

Why graph databases? We'll use an example you've come across….

Examples: Neo4J, Flock, ArangoDB.

Question: You want to find the cheapest flight, regardless of number of stops, from Montreal to Seattle

Flight Data stored as Relational Table

Flight_ID	Start_Airport	End_Airport	Cost
1231	Montreal	Seattle	700
1234	Montreal	Chicago	200
1235	Montreal	Boston	100
2123	Boston	Seattle	400
2124	Boston	Chicago	50
3123	Chicago	Seattle	200
3124	Chicago	Boulder	50
4123	Boulder	Seattle	100
….			

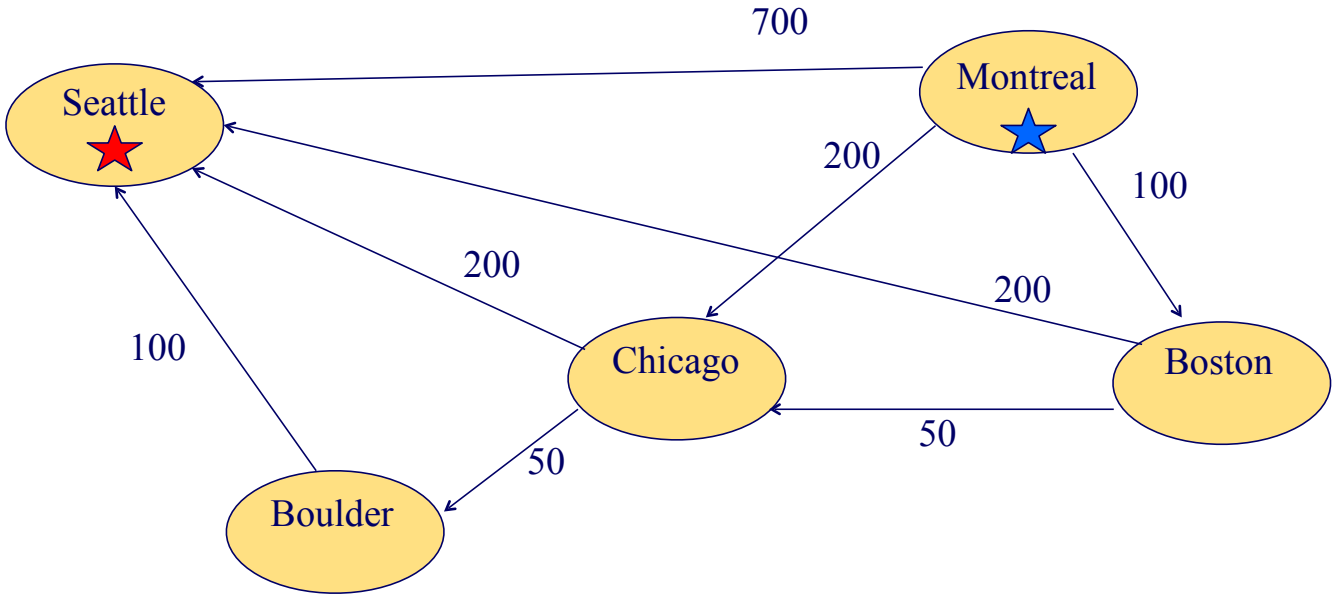
Query for direct flight

```
SELECT Cost
FROM Flights
WHERE Start_Airport='Montreal'
And End_Airport='Seattle';
```

Query for 1-stop flight

```
SELECT (A.Cost + B.Cost)
FROM Flights A,B
WHERE A.Start_Airport='Montreal'
AND A.End_Airport=B.Start_Airport
        B.End_Airport= 'Seattle';
```

An Alternate Data Model



How do you find the cheapest flight plan from Montreal to Seattle ?

- Do you know of any algorithms to do this ?

What is a Graph Database?

A database with an explicit graph structure

Each node knows its adjacent nodes

- As the number of nodes increases, the cost of a local step (or hop) remains the same

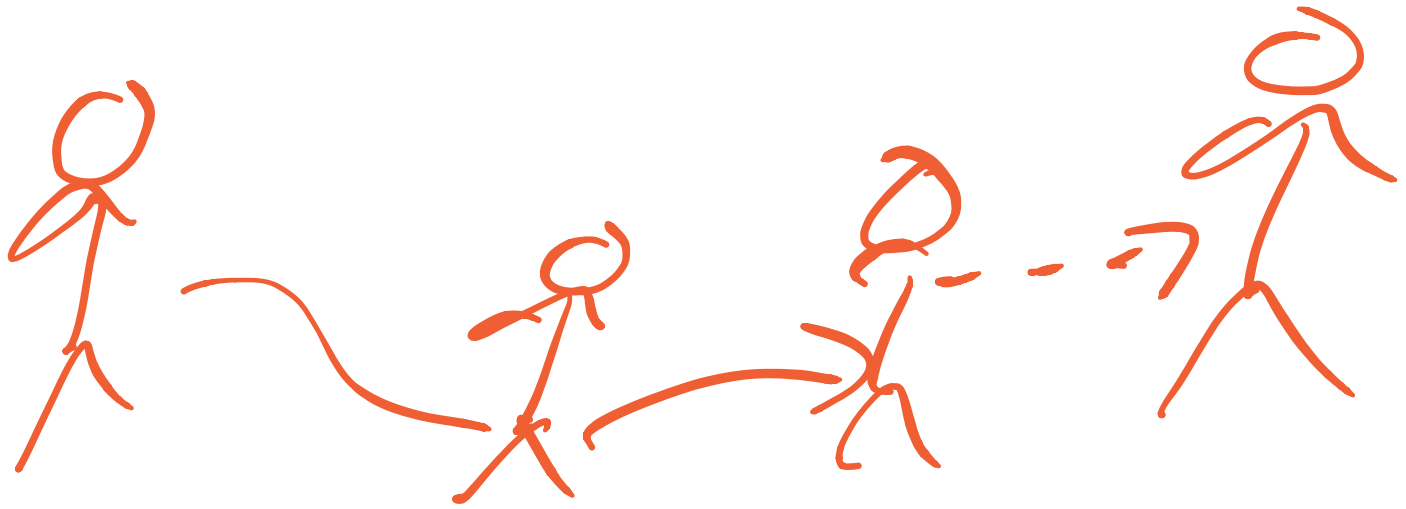
Captures the richness in connectedness of data

- Social network analytics – much easier when modeled as a graph
- Many problems can be represented as graphs (supply chains, transportation, software function call chains, ...)

Graph Examples

Average number of "hops" between two random Twitter users?

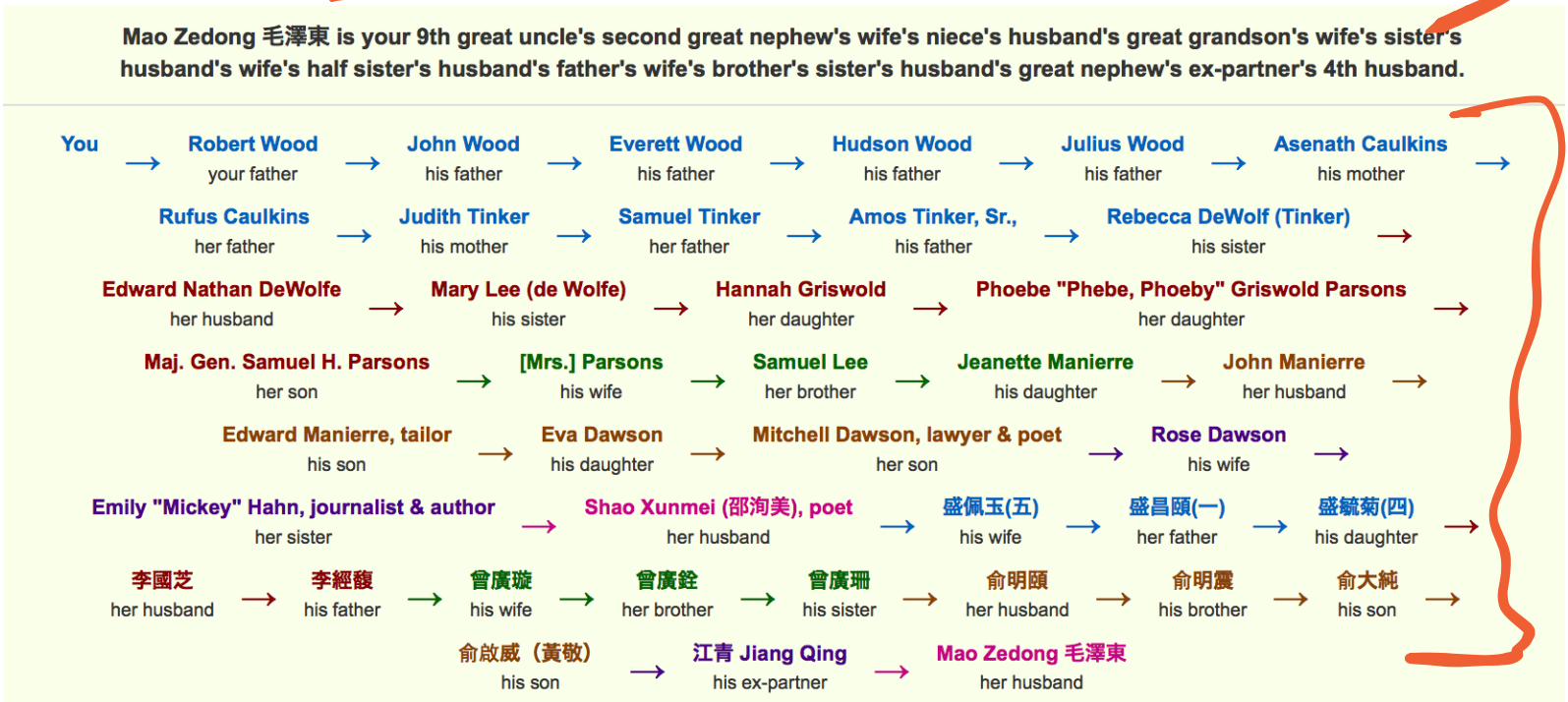
Is Prof. Wood related to....?



Graph Examples

Average number of "hops" between two random Twitter users? **3.43**

Is Prof. Wood related to....?



Should I be using NoSQL Databases?

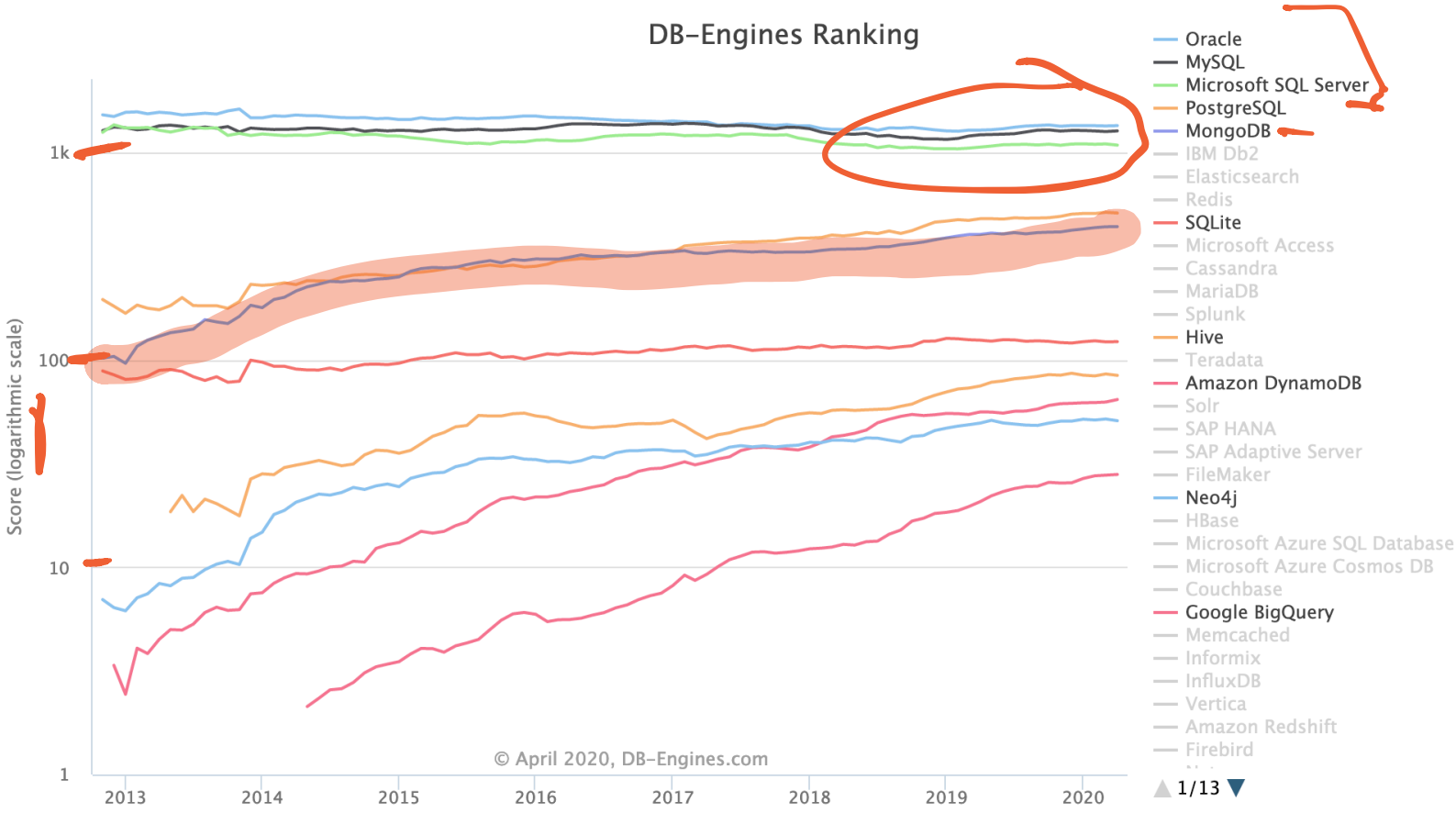
NoSQL Data storage systems makes sense for applications that need to deal with **very very large** semi-structured data

- Social Networking Feeds, Data analytics

For most organizational (ecommerce) databases, which are not that large and have low update/query rates, **regular relational databases are usually the right solution**

- Standards, reliable, ACID

DB Engines Ranking: DBMS systems by popularity



https://db-engines.com/en/ranking_trend