Optimizing Modern Data Processing Systems with Automated Reasoning

Remy Wang

November 30, 2021

University of Washington

Overview

Modern Data Processing

Background: Query Optimization and Reasoning

The Functional Representation of Data

Building Optimizers that Reason

Proposal: Improving Optimizers with Reasoning

Modern Data Processing

Modern Data Processing: Queries

Betweenness centrality of a graph E:

$$C(s,v) = \sum_{t:E(v,t)\wedge D(s,t)=D(s,v+1)} \frac{\sigma(s,v)}{\sigma(s,t)}(1+C(s,t))$$

where D is the distance, σ is the $\#$ of shortest paths.

Modern Data Processing: Queries

Betweenness centrality of a graph E:

$$C(s,v) = \sum_{t:E(v,t) \wedge D(s,t)=D(s,v+1)} \frac{\sigma(s,v)}{\sigma(s,t)}(1 + C(s,t))$$

where D is the distance, σ is the $\#$ of shortest paths.

- Non-relational data

Modern Data Processing: Queries

Betweenness centrality of a graph E:

$$C(s,v) = \sum_{t:E(v,t) \wedge D(s,t)=D(s,v+1)} \frac{\sigma(s,v)}{\sigma(s,t)}(1 + C(s,t))$$

where D is the distance, σ is the $\#$ of shortest paths.

- Non-relational data
- Aggregation & interpreted functions

Modern Data Processing: Queries

Betweenness centrality of a graph E:

$$C(s,v) = \sum_{t:E(v,t) \wedge D(s,t)=D(s,v+1)} \frac{\sigma(s,v)}{\sigma(s,t)}(1 + C(s,t))$$

where D is the distance, σ is the $\#$ of shortest paths.

- Non-relational data
- Aggregation & interpreted functions
- Recursive

Modern Data Processing: Queries

Betweenness centrality of a graph E:

$$C(s,v) = \sum_{t:E(v,t)\wedge D(s,t)=D(s,v+1)} \frac{\sigma(s,v)}{\sigma(s,t)}(1+C(s,t))$$

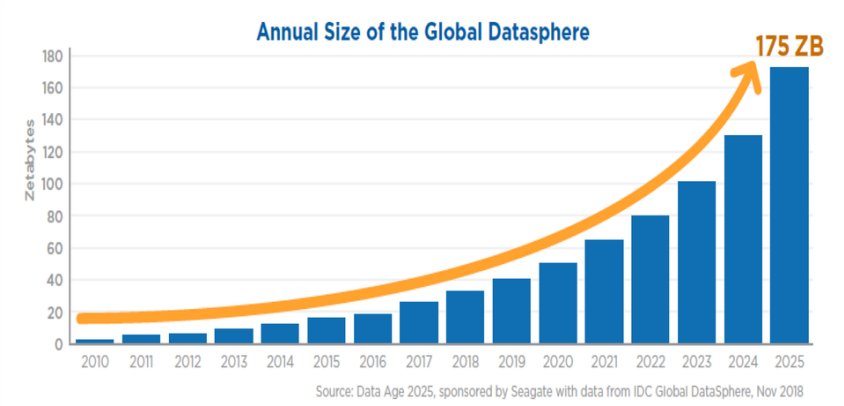
where D is the distance, σ is the $\#$ of shortest paths.

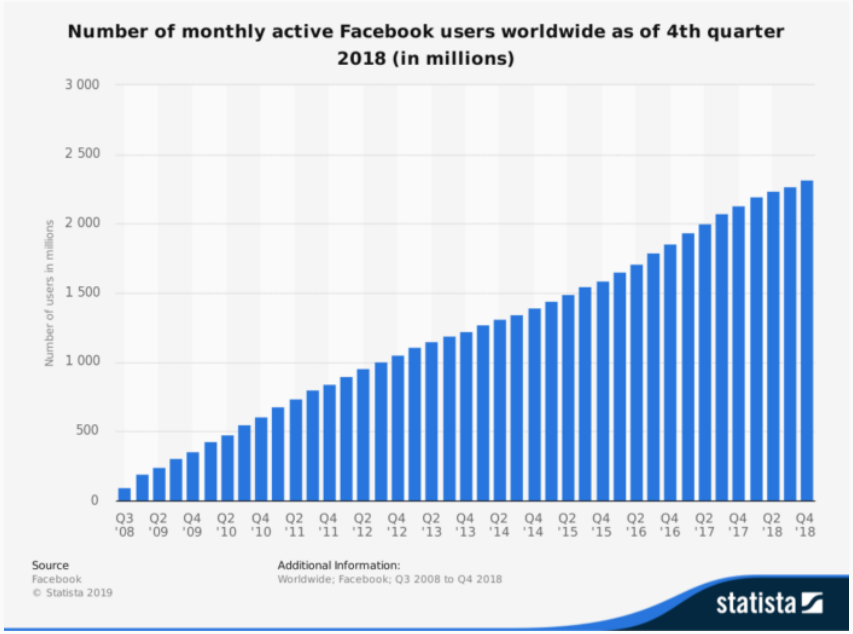
- Non-relational data
- Aggregation & interpreted functions
- Recursive

Expressiveness not well-supported by traditional optimizers:

$$\#[R \times S] \Rightarrow \#[R] \cdot \#[S]$$

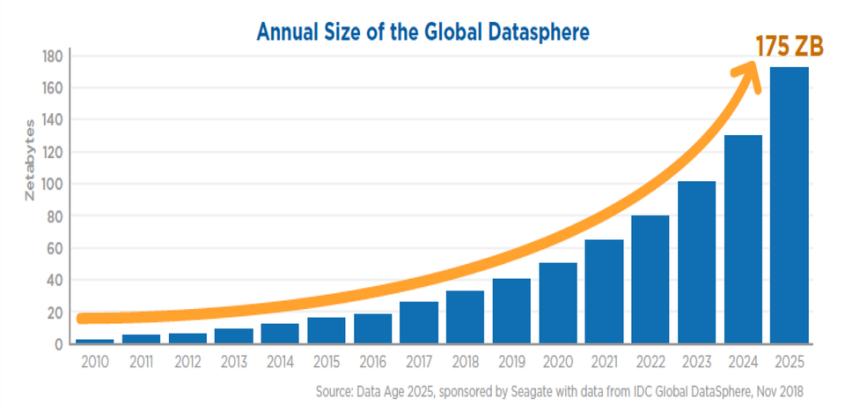
Modern Data Processing: Data

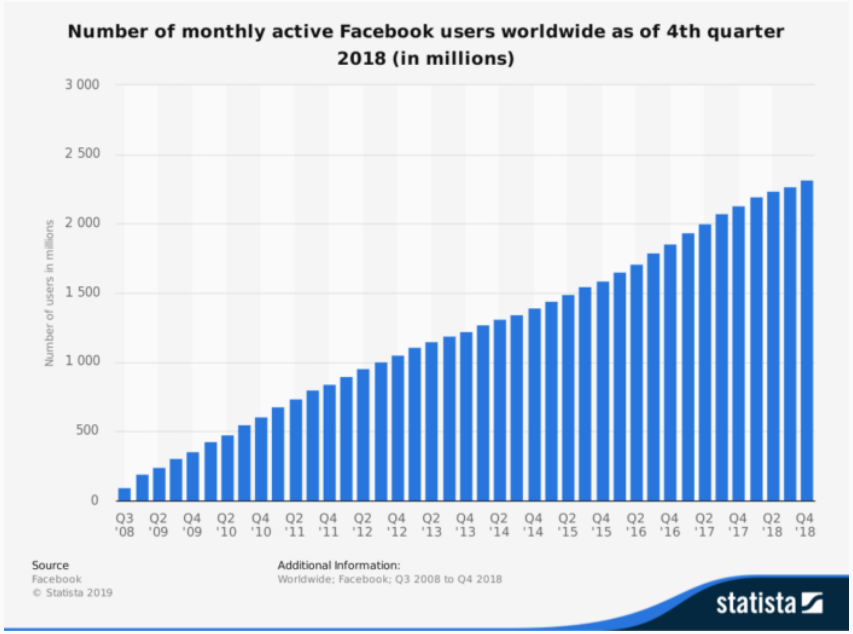




- Data is increasing in **volume** & **velocity**

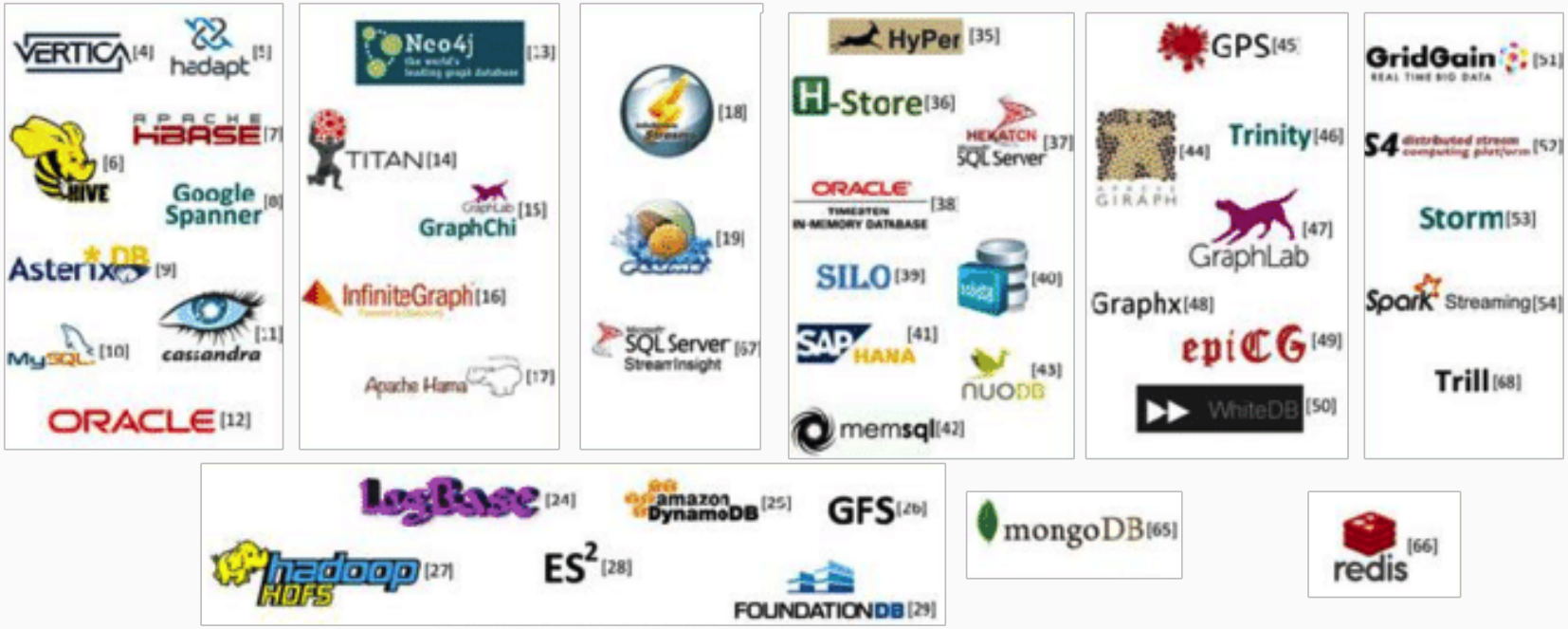
Modern Data Processing: Data





- Data is increasing in **volume** & **velocity**
- The optimizer needs to produce faster plans in shorter time

Modern Data Processing: Systems



- Every data processing system needs an optimizer!
- Simplifying optimizers can save a lot of time for everyone.

My Theses

- Core components and techniques from **automated reasoning** can make query optimizers simpler, more efficient, and more effective.

My Theses

- Core components and techniques from **automated reasoning** can make query optimizers simpler, more efficient, and more effective.
- The **functional representation** of data can enable query optimizers to effectively leverage automated reasoning tools.

Background: Query Optimization and Reasoning

The Functional Representation of Data

Building Optimizers that Reason

Proposal: Improving Optimizers with Reasoning