Team AWFY: Analyzing Massive Social Graphs in a Blink of an Eye

Moritz Kaufmann, Manuel Then, Tobias Mühlbauer, Andrey Gubichev
{kaufmanm,then,muehlbau,gubichev}@in.tum.de

Technische Universität München

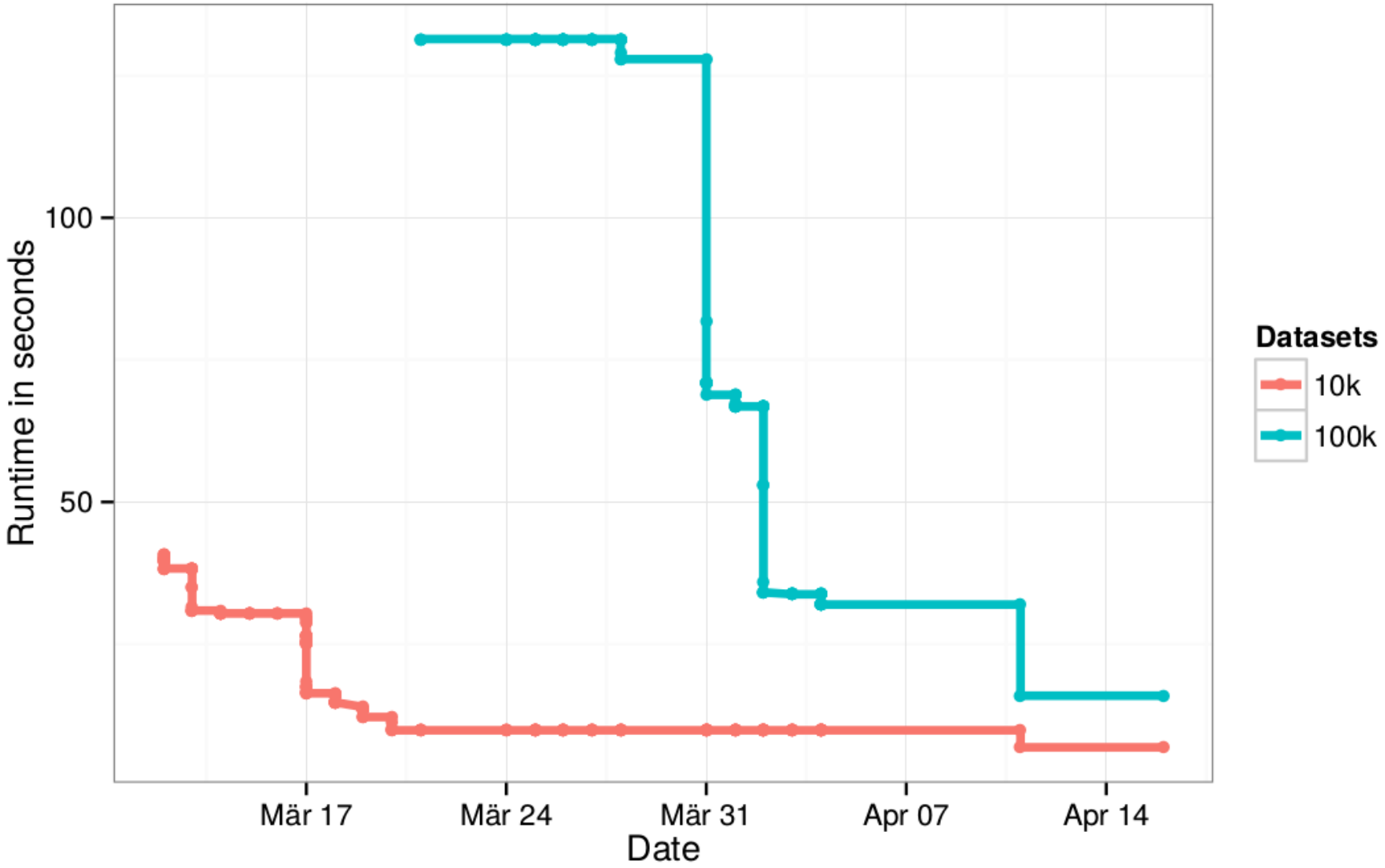
Our Background

- High-performance main memory databases (HyPer)
- Instant loading of text files
- RDF (graph) query execution

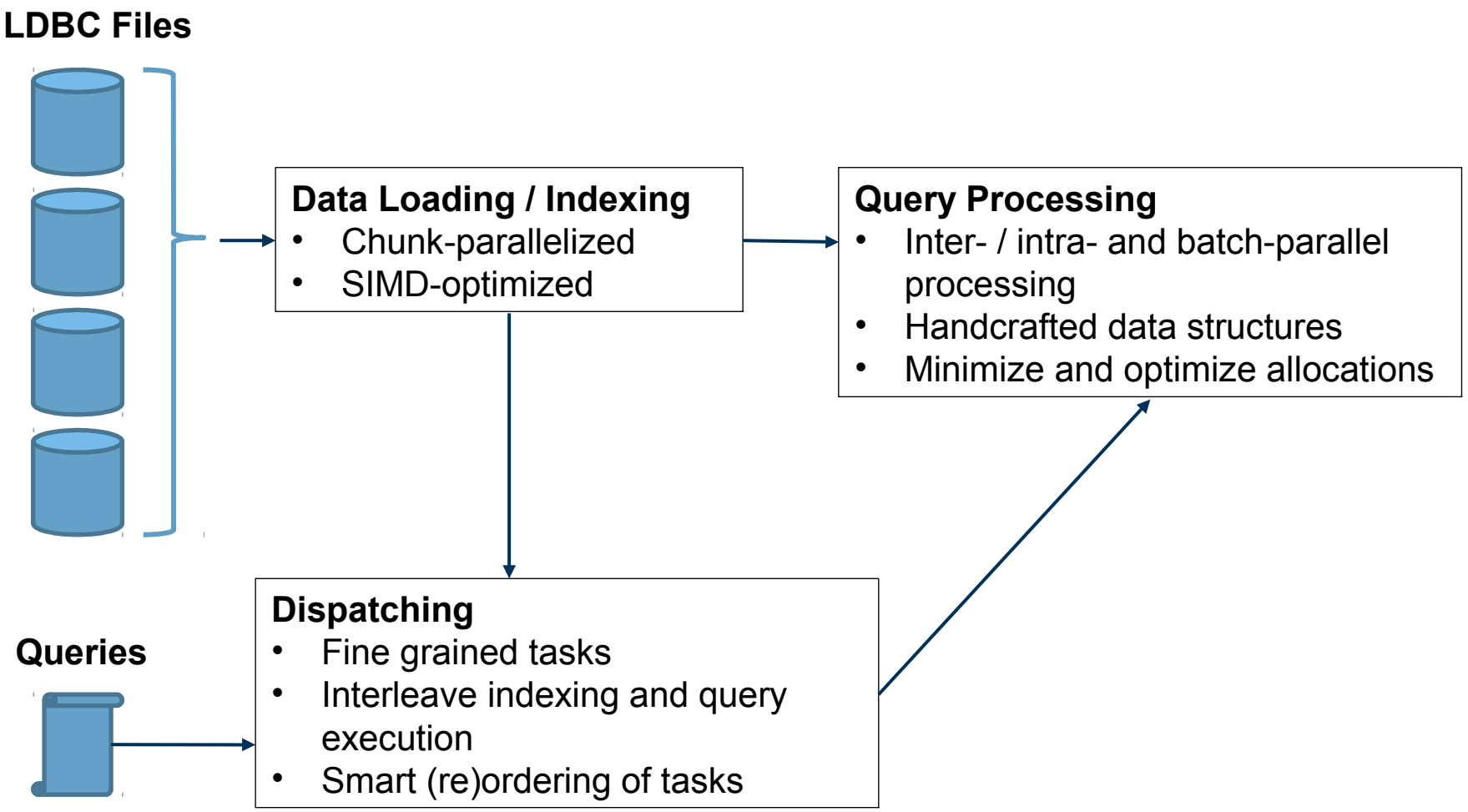
The Challenge

- Shortest Path
- Connected Components
- Region containment
- Graph Centrality (APSP)

Our Runtime over time



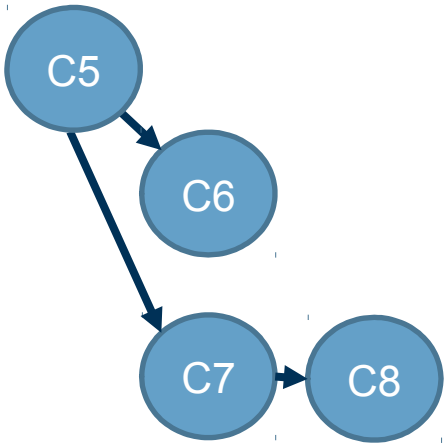
General Overview



Query 1

Task: Find shortest path between persons *a* and *b* over friends that have more than *x* comments in reply to each other

Comment hierarchy

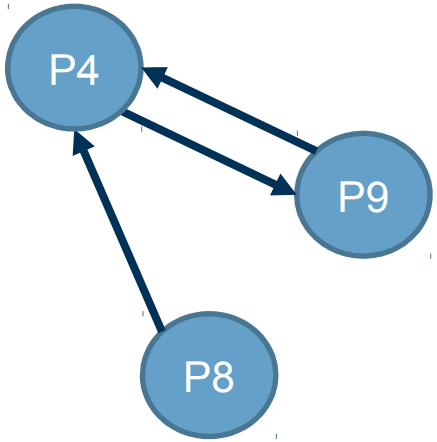


+

Comment authors

Comment	Author
C5	P4
C6	P9
C7	P8
C8	P4

Person communication



SSSP on unweighted graphs → Bidirectional BFS

Query 2

Task: Find *top-k* interests with the largest connected components of persons younger than a specific birthday *b*

Example: k=3, birthday threshold=February / 1995

Current top-k list

Interest	Largest CC
Soccer	9920
Brazil	8137
Hiking	4219

Pruning potential

Interest	Min. Bthdy.	#persons
Travel	~~May / 1994~~	12519
Surfing	May / 1996	~~712~~
Databases	~~June / 1990~~	~~5~~
…	…	…

Query 3

Task: Find the *top-k* pairs of people, that are not more than *h* friends apart, live at place *p,* and have the most interests in common

Basics

- Places are defined as hierarchies
- A person can have multiple places

Solved Challenges

- Fast Set intersection *“SIMD Compression and the Intersection of Sorted Integers”* [Daniel Lemire et al.]
- H-neighborhood → BFS

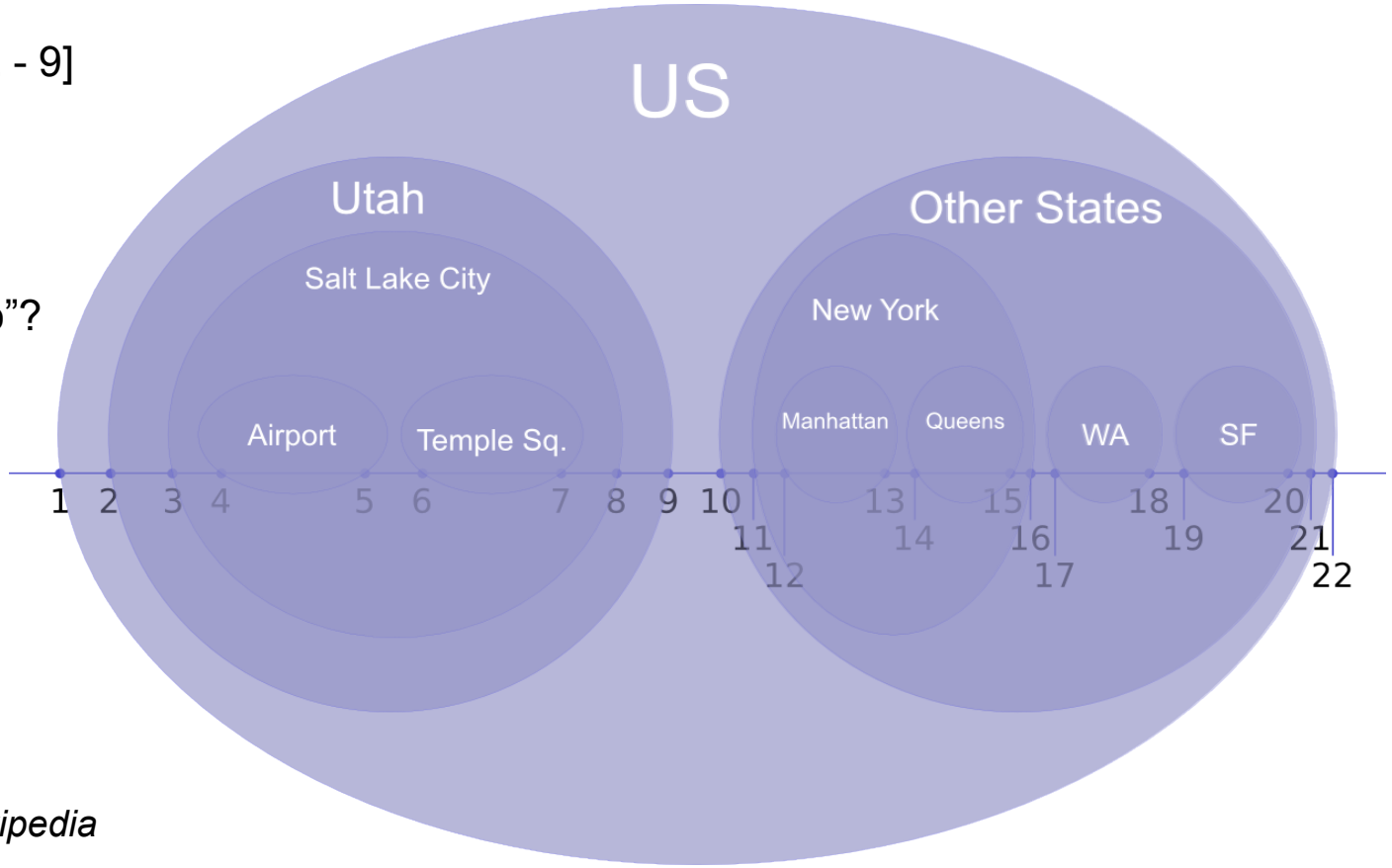
Query 3 (Nested Intervals)

Remaining Challenge: Efficiently check if a person belongs to a place

Reference "Utah" [2 - 9]

Query "Airport"?
[4 - 5]

Query "San Francisco"?
[19 - 20]



Picture adapted from Wikipedia

Query 4

Task: Find *top-k* most “central” people in subgraph defined by query → Naïve solution requires computing APSP

Make it fast → Aggressive Pruning!

- Pruning requires threshold and lower distance sum bound
- Goal A: Get good top-k thresholds early
- Goal B: Have good lower distance bounds for pruning

Solution*: Good lower bound for distance sum*

Query 4 (Distance Sum Estimate)

Intuition: The set of persons someone can reach with $k+1$ hops is the union of the set of persons its friends can reach with k hops.

Set Approximation:

- FM Sketch
- Cardinalities

Conclusion

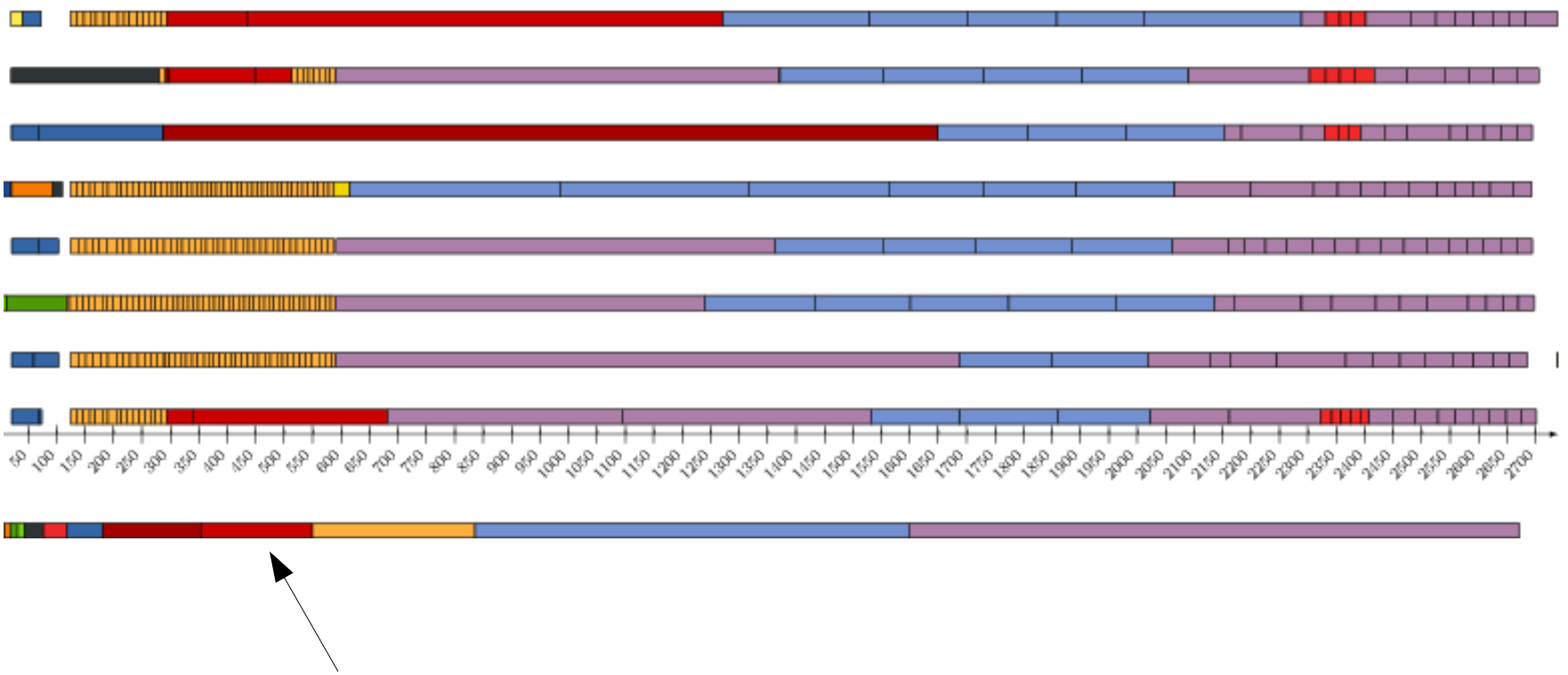
Q4 **dominates** this competition >**75%** of total runtime (in our implementation)

Final performance not predictable

Learned a lot!

Lots of work but even more **fun**!

Runtime Tracing



Relative cost compared to total runtime