University of California, Berkeley
Master of Information and Data Science (MIDS)
W205 – Fundamentals of Data Engineering

Week 8 – NoSQL Graph Databases, Part I

Agenda for Today’s Class

- Attendance and Participation
- Announcements
- Schedule and Due Dates
- Work / Life / School Balance
- Asynch High Level Review in a Nutshell
- Breakouts
- Summary

Attendance and Participation

Please record your attendance and participation for today’s class:

GitHub => ucb_mids_w205_repo => README.md => Attendance and Participation

Announcements

- Upcoming holidays and/or breaks
- Makeup classes for holidays
- Upcoming events
- Student evaluations
- Etc.

Schedule and Due Dates

Take a quick look at the next couple of weeks' due dates:

GitHub => ucb_mids_w205_repo => README.md => Schedule and Due Dates

Work / Life / School Balance

Open Discussion

Student feedback

- About 5 minutes
- How are things going related to work / life / school balance?
- How is w205 going? Difficulty? Time?
- Impact of any natural and/or man-made disasters
- Etc.

Asynch High Level Review in a Nutshell

Each week we will spend about 15 minutes reviewing the most important high level concepts from the asynch

NoSQL Databases

- NoSQL
  - Not only SQL
  - We still try to use SQL or SQL-like query languages if we can!
- Why do people use relational databases when a NoSQL database is a much better fit for a particular case?
  - May not even know that NoSQL databases exist
  - Lack of knowledge about NoSQL databases
  - Inertia – we have always used relational database
  - Corporate standards say we have to use relational database

NoSQL Database Attributes

NoSQL databases typically have one or more of these attributes:

- Key-value
- Document
- In-memory
- Wide column
- Graph
- Time series
- Ledger

Most Common

- We are going to take a deeper look at three of the most common types of NoSQL databases in use:
  - NoSQL graph (week 8 & 9)
  - NoSQL document (week 10)
  - NoSQL key-value (week 10)
- NoSQL wide column is also very common
  - Uses SQL on top
  - User level: very similar to querying SQL against relational table (Since it works like relational database, we won’t cover)

NoSQL Graph Database

- Basic graph
  - Nodes (vertices)
    - Labels for classification purposes
    - Attributes in the form of key-value pairs
  - Relationships (edges)
    - Type for classification purposes
    - Direction
    - Attributes in the form of key-value pairs
- Many problems are naturally represented as graphs
- SQL-like query language

Graph Types and Structures

- Shapes
  - Random, small-world, scale-free
- Characteristics
  - Connected, disconnected, weighted, unweighted, directed, undirected, acyclic, cyclic, trees
- Density
  - Calculation, sparse, dense
- k-partite
  - Monopartite, bipartite, k-partite

Path Algorithms

- Breadth-first search
  - Visit sibling nodes first before visiting child nodes
- Depth-first search
  - Visit child nodes first before visiting sibling nodes
- Eulerian Circuits
  - Visit every relationship only one time
  - Visit nodes one or more times
- Hamiltonian Circuits
  - Visit every node only one time
  - Visit relationships one or more times

Path Algorithms (continued)

- Shortest Path
  - Find the shortest path between two nodes
- A* (A Star)
  - Variation of shortest path, uses heuristic to speed up
- Yen’s
  - Variation of shortest path, finds k shortest paths
- All Pairs Shortest Path
  - Find the shortest path between all pairs of nodes

Path Algorithms (continued)

- Single Source Shortest Path
  - Find the shortest paths from one node pairwise with all other nodes
- Minimum Spanning Tree
  - Find a tree structure path that will visit all nodes in the graph with smallest cost
- Random Walk
  - Given a path size, randomly walk through nodes until we hit the path size

Breakouts

GitHub => ucb_mids_w205_repo => breakouts

(time permitting, we may not get to all of them)

Summary

Instructor will give a brief (about 2 minute) summary of today’s class.