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

Week 10 – NoSQL Key-Value Databases

Agenda for Today’s Class

- Attendance and Participation
- Announcements
- Schedule and Due Dates
- Work / Life / School Balance
- Asynch High Level Review in a Nutshell
- Project 3
- 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 Document Database

- Collection of documents
  - Originally XML
  - Then JSON
  - Now JSON-like
- Key – Value
  - Key’s value is the document
- Queries return entire documents instead of rows
- Advantage – JSON-like document can hold the equivalent of several SQL relational database tables

Refresher (Transactional / Analytical)

- Transactional Databases
  - Execute the business
  - Normalized, 3NF, Third Normal Form, no duplication of data
- Analytical Databases
  - Analyze the execution of the business
  - Denormalized – data is duplicated to make analytics more convenient
- NoSQL Document Databases
  - If used as an Analytical Database, we will want to duplicate data to make analytics more convenient

Multiple POVs (Points of View)

- We may want to analyze data from several different POVs
  - Store, customer, product, quarter, month, etc.
- Create a separate document for each POV
- For each analysis, select the POV that is most convenient
- Denormalized
  - Data is duplicated
  - Disadvantage: updates have to update multiple copies
  - Refresh frequency: every X minutes, every X hours, once a day, once a week, etc.

NoSQL In-Memory Key-Value Databases

- In-Memory
  - Entire database is stored in memory and written at a periodic interval to disk – must fit in memory
- Key-Value
  - Update logic similar to Python dictionary
  - If key does not exist, add the key-value
  - If key exists, overwrite the value with new value

NoSQL In-Memory Key-Value Databases (cont’d)

- Value
  - Can be any binary data – no required format
  - Can be JSON
  - When value is JSON, somewhat similar to NoSQL Document Database (will compare on the next slide)
- Queries
  - By key – extremely fast
  - Not by key – exhaustive search

NoSQL Key-Value vs. NoSQL Document

- NoSQL key-value (assume in-memory)
  - Good for databases that can fit in memory (or memory budget)
  - Typical query is by key
  - Non-key queries are carefully considered
  - Need it as fast as possible
- NoSQL document
  - Good for databases that are too big for memory (or budget)
  - Databases that could fit in memory but have a lot of non-key queries

Project 3

GitHub => ucb_mids_w205_repo => projects => project_3

- Videos going over project 3 are provided, so we won’t spend class time going over it today
- First breakout today will be related to project 3

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.