CSL7090
Software & Data Engineering

Lecture #16 NoSQL, Graph, XML, Key-Value Store

Course Instructors:

Sumit Kalra, Assistant Professor, IIT Jodhpur
Ram Subramaniam, Co-founder - My Money Karma

Google Classroom Code: cezptvv

Limitations of Relational Databases

Inadequate representation of data

Semantic Overloading

Weak support for recursion

Homogeneous structure of all the records - rigid schema

Short-lived transactions

Non-versioned data

Additional Challenges

Complexity

Schema Independence

Sparseness

Self-descriptiveness

Variability

Scalability

Volume

NOSQL

Not Only SQL

Data models others than conventional relational schema

Support flexible schema structure

May not facilitate ACID (in particular Consistency)

Graph Databases

Addresses Semantic Overloading problem of relational databases

Entities - Nodes/Vertices

Relations - Arcs/Edges

Can store information in Nodes as well as in Edges

Eg. Social Networks, Geographical Information

Types: Simple undirected graph, simple directed graph, undirected multigraph, directed multigraph, weighted graphs

Traversal: Eulerian Path/Cycle, Hamiltonian Path/Cycle

Graph Data Structures

Edge List

Adjacency Matrix

Incidence Matrix

Adjacency List

Incidence List

Some more pointers on graph databases

Property Graph Model

- A multi-relational graph, Nodes and Edges can be of different types

Advanced Graph Models

- Hypergraph, Nested graph

Example Systems

- Apache TinkerPop, Neo4J, HyperGraphDB, and many more . . .

XML

eXtensible Markup Language

Standards: XML Schema, XQuery, XSLT

Semi-structured Schema, tree shape

XML Document, Document Type Definition, XML Schema Definition, XML Parsers

XML Query Languages: XPath, XQuery (FLOWER expression: FOR, LET, WHERE, ORDER BY, RETURN. FLWOR is loosely analogous to SQL's SELECT-FROM-WHERE)

XSLT - eXtensible Stylesheet Language Transformation

XML Concurrency Control: Node-locking, Path-Index-Locking

Example Systems: eXistDB, BaseX, and manay more

Key-Value Store

Schemaless database

A powerful framework for scalable and distributed processing: Map-Reduce

4 basic operations:

- Split: split key-value pairs in disjunct subsets
- Map: execute map function on key-value pairs, outputs intermediate k-v pairs
- Shuffle: group intermediate k-v pairs based on key
- Reduce: reduce value for each group to usually one key-value pair

Map-Reduce Example

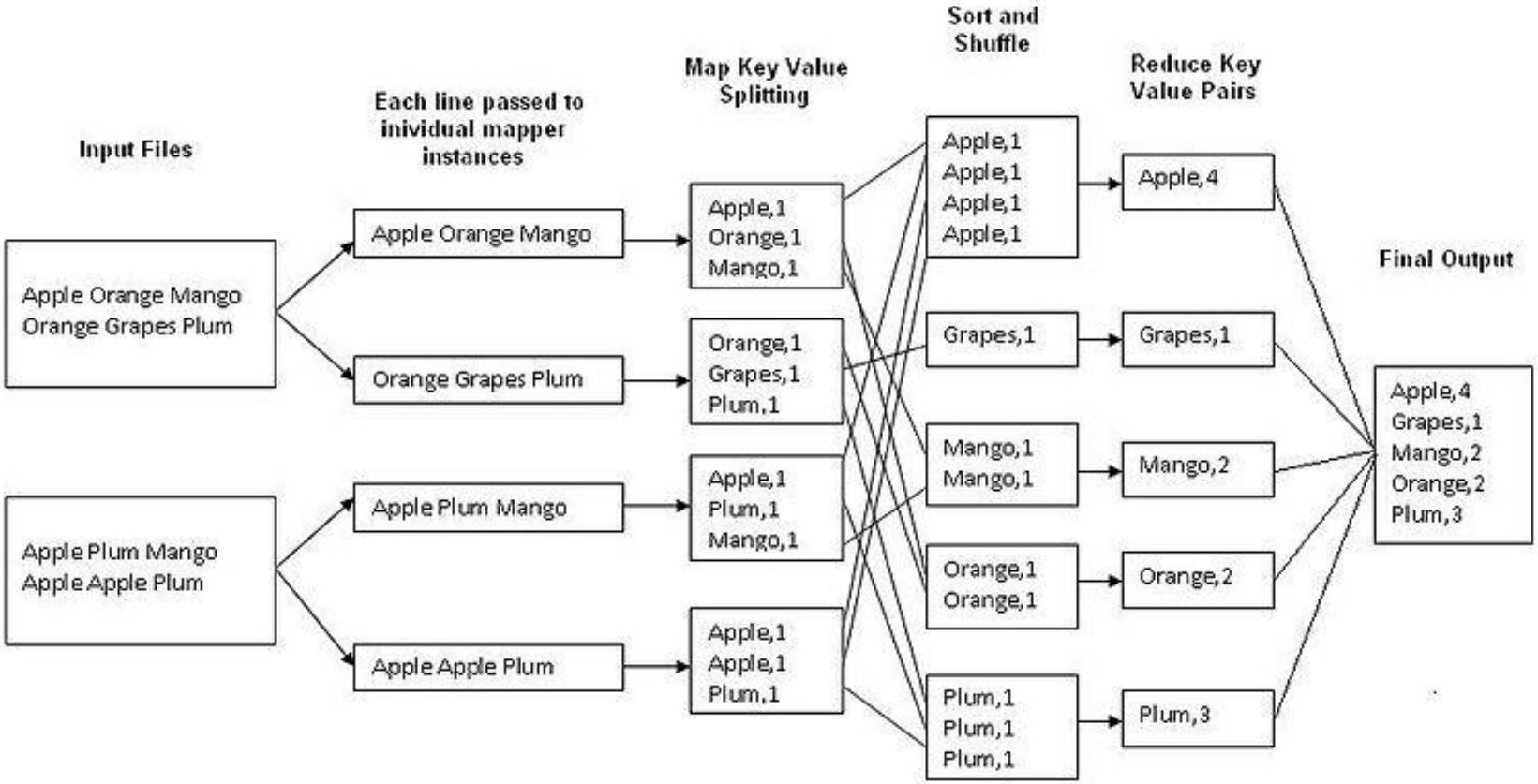
Counting words in a document:

Split - Input: document, Output: sentences

Map - Input: sencetens, Output: (word, 1)

Shuffle - Input: (word, 1), Output: (word, 1:1:1:1…)

Reduce - Input: (word, 1:1:1:1…), Output: (word, SUM(1:1:1:1…))


Input Files
Each line passed to individual mapper instances
Map Key Value Splitting
Sort and Shuffle
Reduce Key Value Pairs
Final Output
Apple Orange Mango Orange Grapes Plum
Apple Plum Mango Apple Apple Plum
Apple Orange Mango
Orange Grapes Plum
Apple Plum Mango
Apple Apple Plum
Apple,1 Orange,1 Mango,1
Orange,1 Grapes,1 Plum,1
Apple,1 Plum,1 Mango,1
Apple,1 Apple,1 Plum,1
Apple,1 Apple,1 Apple,1 Apple,1
Grapes,1
Mango,1 Mango,1
Orange,1 Orange,1
Plum,1 Plum,1 Plum,1
Apple,4
Grapes,1
Mango,2
Orange,2
Plum,3
Apple,4 Grapes,1 Mango,2 Orange,2 Plum,3

Any Questions?

Homework:

Try transferring data from SQL to NoSQL system and vice-versa

Next Class:

Setting up DB on cloud. Flask API framework for Python - Dr. Ram

Contact: sumitk@iitj.ac.in