Final Review

** This is a review of some post-midterm topics.
This review is not exhaustive.
You are responsible for covering the entire course.
The final will be comprehensive, with more weightage on post-midterm topics **

Anurag Nagar

Big Data Class

Outline

Topics Covered Post-Midterm

List of topics covered post-midterm:

- Spark GraphX / GraphFrames
- Structured Streaming
- Hive and Impala
- NoSQL technologies
- MongoDB
- HBase
- Cassandra

Outline

Spark GraphFrames

What is GraphX?

- Unifies traditional computing and graph based computing.
- Can read tabular data, run graph algorithms, and save data as graph or table.
- Graph computation is everywhere - PageRank, Hyperlink analysis, Term-Document graph, Community Detection, Topic Modeling, etc
- Rather than GraphX, which is RDD based, we worked with GraphFrames, which are DataFrame based.

Questions

GraphFrames are contained in which library?

Questions

GraphFrames are contained in which library?

org.graphframes.GraphFrame

Questions

When instantiating a GraphFrame object, two DataFrames are needed. Explain?

Questions

When instantiating a GraphFrame object, two DataFrames are needed. Explain?

First DataFrame should contain data about vertices (nodes) and their properties.
Second DataFrame should contain data about the edges and their properties.

Questions

What are the columns required in each DataFrame?

Questions

What are the columns required in each DataFrame?

Vertices DataFrame: Id, vertex property
Edges DataFrame: sourceid, destinationid, edge property

Questions

What are the columns required in each DataFrame?

Questions

What are the columns required in each DataFrame?

Vertices DataFrame: id, vertex property
Edges DataFrame: sourceid, destinationid, edge property

Questions

If **g** is a GraphFrame object, which of the following are valid methods that can be called on it?

1. g.vertices.show(...)
2. g.edges.show(...)
3. g.find(pattern)
4. g.filterVertices(criteria)
5. g.connectedComponents.run()
6. g.stronglyConnectedComponents.run(...)
7. g.pageRank(...)
8. g.triangleCount.run()

Questions

If **g** is a GraphFrame object, which of the following are valid methods that can be called on it?

1. g.vertices.show(...)
2. g.edges.show(...)
3. g.find(pattern)
4. g.filterVertices(criteria)
5. g.connectedComponents.run()
6. g.stronglyConnectedComponents.run(...)
7. g.pageRank(...)
8. g.triangleCount.run()

All of the above. See https://graphframes.github.io/graphframes/docs/_site/user-guide.html

Review

Go through the examples for above topics from class lab and quiz.
Remember how to run each i.e. parameters, and what they mean.

Outline

Structured Streaming

Idea: Run streaming queries just like you would run static queries.
System takes are of updating results periodically, making it fault tolerant, handles out of time data, watermarking.

Structured Streaming

Structured streaming can read from a variety of sources and can write to various sinks.
E.g. Kafka, file system, etc

Structured Streaming

How does Structured Streaming store the streaming data?

Structured Streaming

How does Structured Streaming store the streaming data?

Unbounded table. New rows appended to the table
For details see

https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

Structured Streaming

How does Structured Streaming achieve fault tolerance?

Structured Streaming

How does Structured Streaming achieve fault tolerance?

Checkpointing

Structured Streaming

At each trigger point, how does the system write its external output?

Structured Streaming

At each trigger point, how does the system write its external output?

One of three modes: Complete, Append, Update

Structured Streaming

Read about event time, handling late data, watermarking

Outline

Hive and Impala

Understand following for Hive:

- Architecture of Hive
- Hive partitions, and partition keys
- Practice Hive queries

Hive and Impala

Understand following for Impala:

- Architecture
- Daemon processes and their roles
- How is Impala so fast
- Practice queries

Outline

NoSQL technologies

Understand following:

- Why strict ACID is difficult to achieve for distributed and partitioned data
- For Big Data, BASE is more useful than ACID
- Understand eventual consistency
- Understand CAP theorem and which database is where on the CAP axis.
- Types of NoSQL databases: Key-Value stores, Document Databases, Column Oriented Databases and Peer-to-Peer databases.

Outline

MongoDB

Understand following:

- MongoDB hierarchy - databases, collections, documents, fields
- Basic MongoDB query syntax: db.table.find(...), db.table.aggregate(...), db.table.mapReduce(...)

Outline

HBase

Understand following:

- Idea of a column oriented database, column family, columns, versioned cells
- It's an architecture on top of HDFS that provides fast random read and writes, rather than bulk read/write provided by HDFS
- Data always ordered by row key
- Regions and RegionServers
- Basic query syntax

Outline

Cassandra

Understand following:

- Properties - P2P, linearly scalable
- Architecture - idea of coordinator for a request, Replication Factor (onto how many nodes should the coordinator *send* a write request)
- Write Consistency Level (how many nodes must *acknowledge and write* the write request of coordinator)
- Read Consistency Level (how many nodes must *acknowledge and reply* with their timestamped data.
- Types of consistencies: Any, One, Quorum, All. Which provides fastest response, which guarantees no stale read, which guarantees absolute consistency

Cassandra

Understand following:

- Partitions, Partition Keys, Hashing, Token range
- First copy of replica stored to node that owns that token range. e.g. if node X owns token range from 0 - 24, then any data whose hash value falls in that range will be stored in node X as a primary copy.
- How is network topology shared among peers? Gossip Protocol

Cassandra

Understand following:

- Data model, rows as column family, columns as key-value pairs, data stored according to column key
- Partition key determines how data is stored, Clustering key is an index within a partition
- Partition key + Clustering key form the primary key
- What predicates have to be specified in the WHERE clause? At least Partition key
- Go through lab examples