CS4225/CS5425 Big Data Systems for Data Science

Exam

Ai Xin
School of Computing
National University of Singapore
aixin@comp.nus.edu.sg

Exam

- Date: Nov 29, Wednesday
- Time: 13:00pm – 15:00pm
  - Students are allowed to enter the venue at 12:50pm
  - Students will not be permitted to enter the venue after 14:00pm
- Venue: MPSH2-A
  - NUS Multipurpose Sports Hall 2 (MPSH2)
- F2F - Hardcopy (pen and paper)
- Open Book Exam
  - Any physical materials are allowed
  - Calculator is allowed
  - Any other electronics devices are NOT allowed

Exam

- Focus is on understanding and application, not facts / memorization
- Question structures (total 50 marks):
  - Ture / False question with a short explanation / justification
  - Application / Scenario Based Question
    - Give you a practical scenario and let you come out a solution / suggestion
- Example questions
  - Integrative: Require you to combine knowledge from different weeks of content
  - "Application": Require you to apply your knowledge of fundamental concepts to reasonably practical scenarios.
  - "Why not": Example, Tommy proposed a solution A to solve problem B. Tell me what is the problem with solution A and how to overcome this problem

Scope of Exam

- **Scope**: the content in the lectures (1-10) and tutorials(1-5)
- **Out of scope**:
  - The content marked as “optional” or in the appendix
  - Additional information in the comment box
    - Some notes in the comment box is explaining / clarifying the content in the slides, which is not additional information.
- In the following, I will
  - Have a revision on the **key points** that we learnt after recess week.

Spark I

- Introduction
  - In memory processing and easy to use
  - Driver and Executors
- RDD
  - Distributed, Immutable, Lazy Transformations, Action to trigger the computation
  - Caching an RDD: when it is expensive to compute and needs to be re-used many times
- DAGs
  - The lineage of an RDD, Within Stage (Narrow Transformation), Across Stage (Wide Transformation)
- DataFrame: the recommended interface
  - filter, sort, join, groupby, and etc.
- Datasets: type-safe during compile time

Spark II

- Spark SQL and Catalyst Optimizer
  - Unifies Spark components and permit various languages
  - Tell Spark what to do and then Spark will generate an optimized plan
- Machine Learning Pipeline
  - Pre-process the Data
  - Build the model using Training Data
  - Evaluate the Model using Testing Data
- Implementing ML Pipeline using Spark Mllib
  - Transformer
    - transform() method: map df1 to df2
  - Estimator
    - fit() method: takes in data and outputs a fitted model (“transformer”)
  - Model training stage: iterative distributed in-memory computation
    - Cache training data in memory across iterations
    - Use broadcast variable to save & broadcast weights iteration by iteration

Stream

- Spark: Structured Streaming (latency of a few seconds)
    - Micro-Batch approach, Incremental Execution
    - Five Steps to define a streaming query
    - Stateless Transformation: filter(), map(), etc.
    - Stateful Transformation / Aggregation
        - Not Based on Time: groupBy().count()
        - Based on Time:
            - Event time vs. Processing time,
            - Tumbling Windows, Overlapping / Sliding Windows
            - Watermark: handling late data

- Flink: Real-time streaming processing (latency of milliseconds)
    - a distributed system for stateful parallel data stream processing
    - Event time processing with watermarks
    - State Management: distributed snapshots using checkpoint barriers

Graph

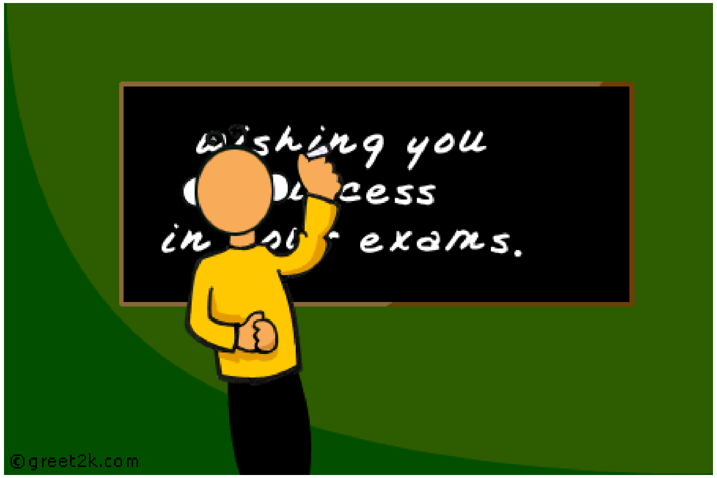
- Simplified PageRank

$$r_j^{(t+1)} = \sum_{i\to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$ or equivalently $$r = Mr$$

  - Flow formulation
  - Random Walker formulation
- PageRank with Teleports
  - Flow Equation: $r_j = \sum_{i\to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$
  - Google Matrix
    - $\boldsymbol{r} = \boldsymbol{A} \cdot \boldsymbol{r}$
    - $A = \beta\, M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}$
- Topic Specific PageRank
  - A topic-specific set of “relevant” pages (teleport set)
- PageRank Implementation
  - Pregel Model, Think like a Vertex, superstep, compute()
  - Partition (Edge Cut) and assign to Workers

Evolution of Data Architectures

- Relational Database
  - strong transactional ACID guarantees
- Data Warehouses
  - a central relational repository, ACID guarantees
- Data Lake
  - A distributed storage solution, runs on commodity hardware and easily scales out horizontally
  - Decouples the distributed storage and computing
  - Mostly cannot provide ACID guarantees, lack of schema enforcement
- Lakehouses: data lake + data warehouse
  - Flexible, low cost, scale + ACID transactions
  - Delta Lake solution: DeltaLog (a single source of truth).


wishing you
cess
in
exams.
©greet2k.com

CHIBIRD
study bunnies

chibird