CS460
Systems for Data Management and Data Science

Introduction to Course Project

Logistics

- **Project registration (Moodle): 27th Feb, 2023 23:59**
- Project Deadline: 19th May, 2023 23:59
- Access to repositories: ~ 6th March, 2023 (skeleton available earlier)
- Graded automatically with tests
  - Only write code in `src/main/scala/app`
  - Last commit before deadline on `main` branch will be graded
- Project IDE: IntelliJ Idea
  - Free community edition
  - ultimate edition on academic license (epfl.ch email)
- Programming Language: Scala

Learning Goals

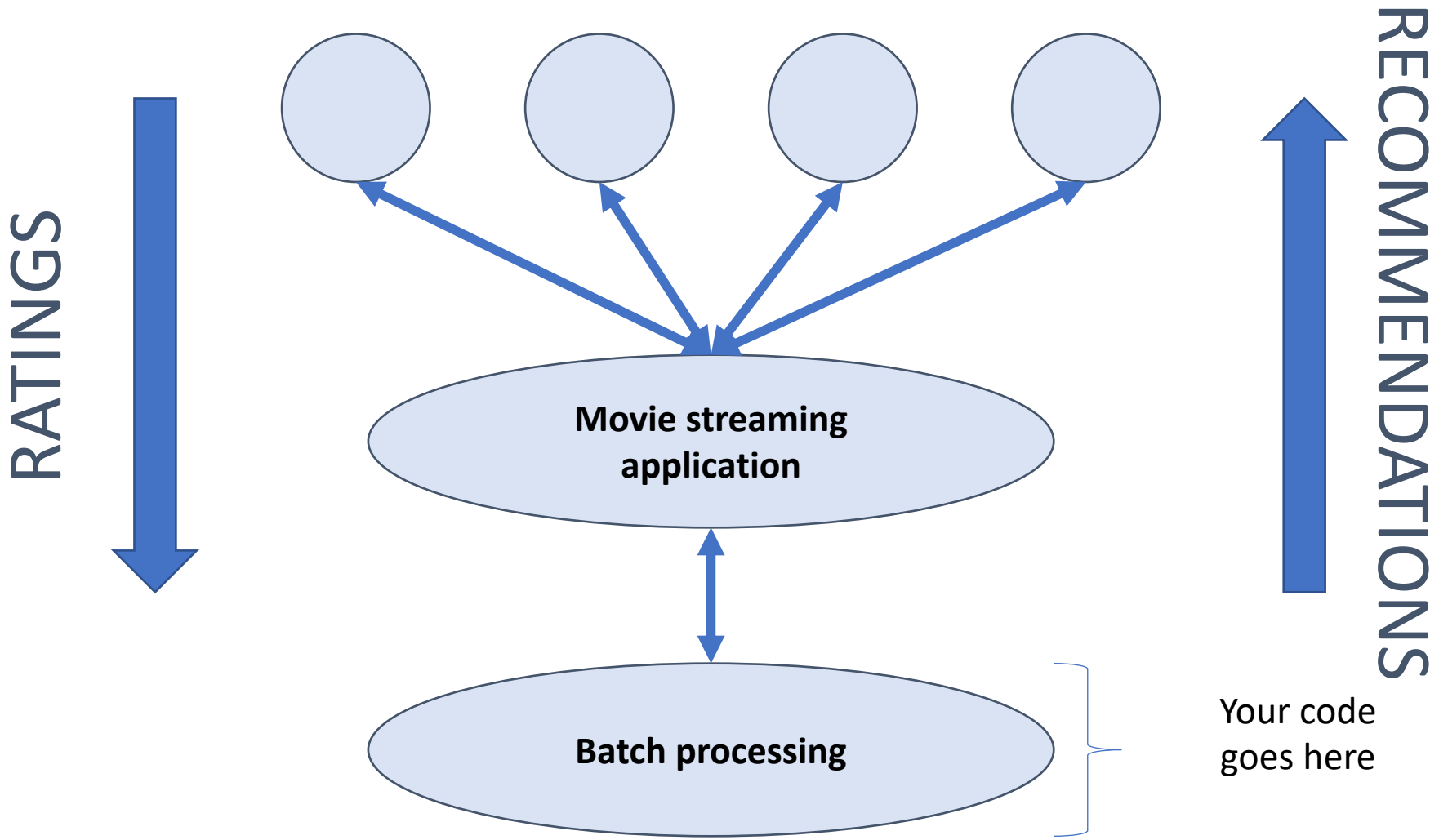
- Apache Spark
  - Unified engine for large-scale data analytics/science and machine learning

- Data Management / Data Science concepts
  - Loading / Caching/ Pre-processing
  - Data partitioning
  - Data shuffling vs broadcast
  - Answer analytical questions
  - Predictive analytics / Recommender systems / Machine learning
  - …

Project Highlights

- Three milestones (single-deadline for all)
  1. Analyzing data with Apache Spark
     - Data loading & Simple data analysis
  2. Movie-ratings pipeline
     - Aggregations & Incremental maintenance
  3. Prediction serving (recommender system)
     - Similarity based recommender: Locality-Sensitive Hashing & Collaborative Filtering
- Dataset: `Movielens`
  - Three sizes:
    - Small for development/debugging
    - Medium for testing/ automatic-testing on Gitlab
    - Large for hands-on experience with cluster

The Usecase


RATINGS
Movie streaming application
Batch processing
RECOMMENDATIONS
Your code goes here

The Data Processing Pipelines

- MovieLens data + simulated ratings
- Answering analytical questions about data (milestone 1)
- From user ratings to average ratings (milestone 2)
  - Average ratings from log
  - Statistics for movies in specific genres
  - Updates on log propagated to average ratings
- From movie keywords to recommendations (milestone 3)
  - LSH: Similarity-search based on keywords
  - Collaborative filtering through spark mllib

You will not have to implement a full system, just the functionality in isolation

Project description & Skeleton

Milestone details

Final Remarks

- You need to fill non-implemented code in the provided skeleton program
- You can run tests locally with Intellij (`src/main/test`)
- IMPORTANT: Do not edit build files / interfaces
  - Auto-grader will fail if you change any interface definition as in skeleton
- Only latest commit in the main branch will be graded
- More info on running on cluster later in March/April