CS 378 – Big Data Programming

Lecture 13

more on

Data Organization Patterns

Review

- Assignment 6 – User Session

- Questions/Issues?
  - How to determine VDP (detail page) versus SRP (results page)

  - Count for |action:click| = 1047
  - Count for |type:action| = 867
  - What's going on here?

Assignment 6

- Define an Avro object for user session
  - One user session for each unique userID/apikey
  - Session will include an array of impressions
  - Impressions ordered by timestamp
  - Each impression will contain an array of IDs (0 or more)

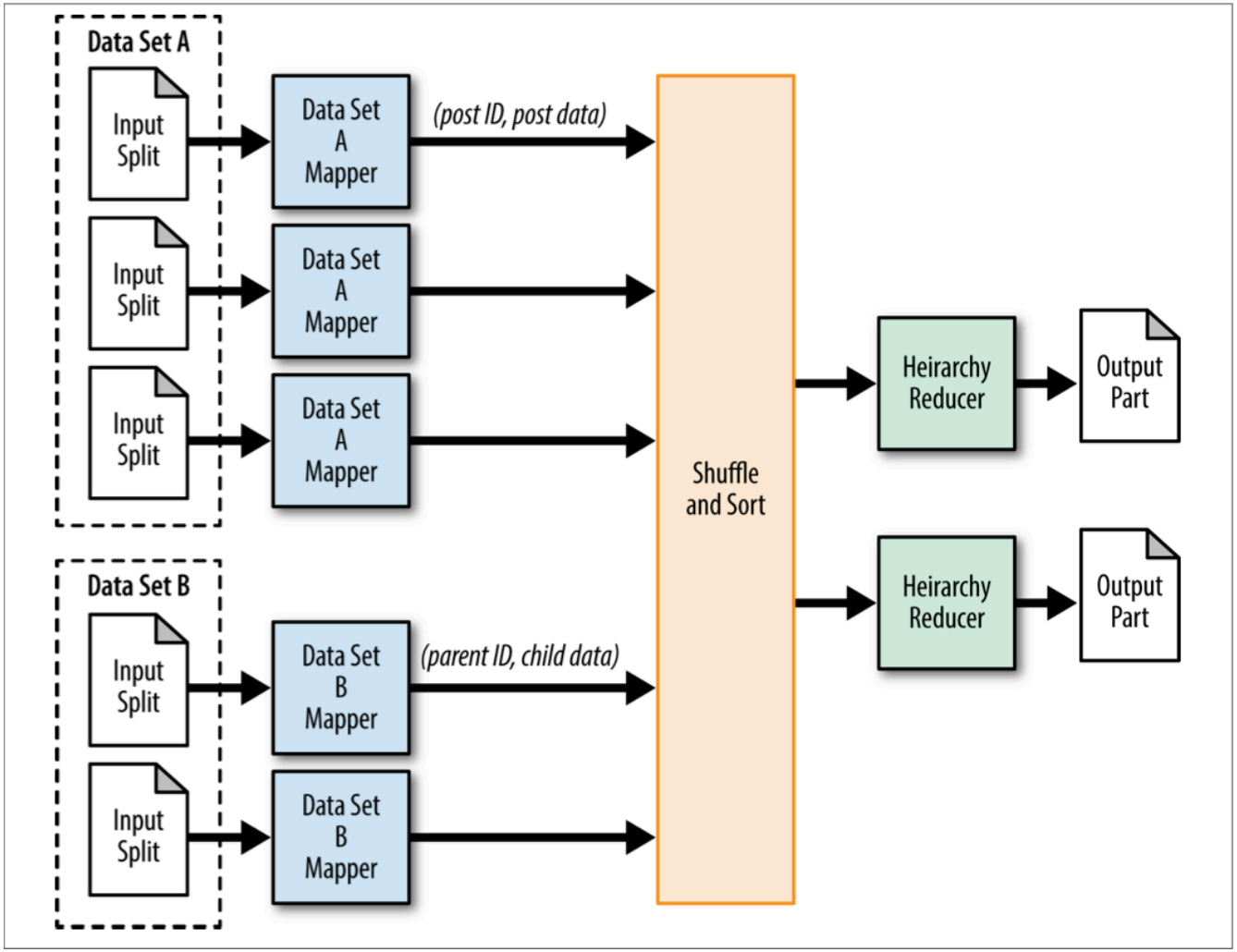
- Identify data associated with the session as a whole
- Identify data associated with individual impressions
- Include all the fields listed in the assignment
- Create enums where requested

Data Organization Patterns

- Structured to hierarchical pattern
  - User session is one such example
    - Organizing web logs by user
  - Textbook shows organizing posts and comments from StackOverflow

Data Flow

Figure 4-1 from MapReduce Design Patterns



Partitioning

- Organize “similar” records into partitions

- Why?
  - Future jobs will only focus on subsets of the data
- Partitioning schemes:
  - Time: hour, day, week, month, year
  - Geography: ZIP, DMA, state, time zone, country
  - Data source: web site
  - Data type

Partitioning

- No downside, as a mapReduce job can run over all partitions if needed

- We do need to know *a priori* how many partitions we want
  - Can run a job that scans and summarizes the data
  - Get possible values, and counts
  - Just like we did for user sessions

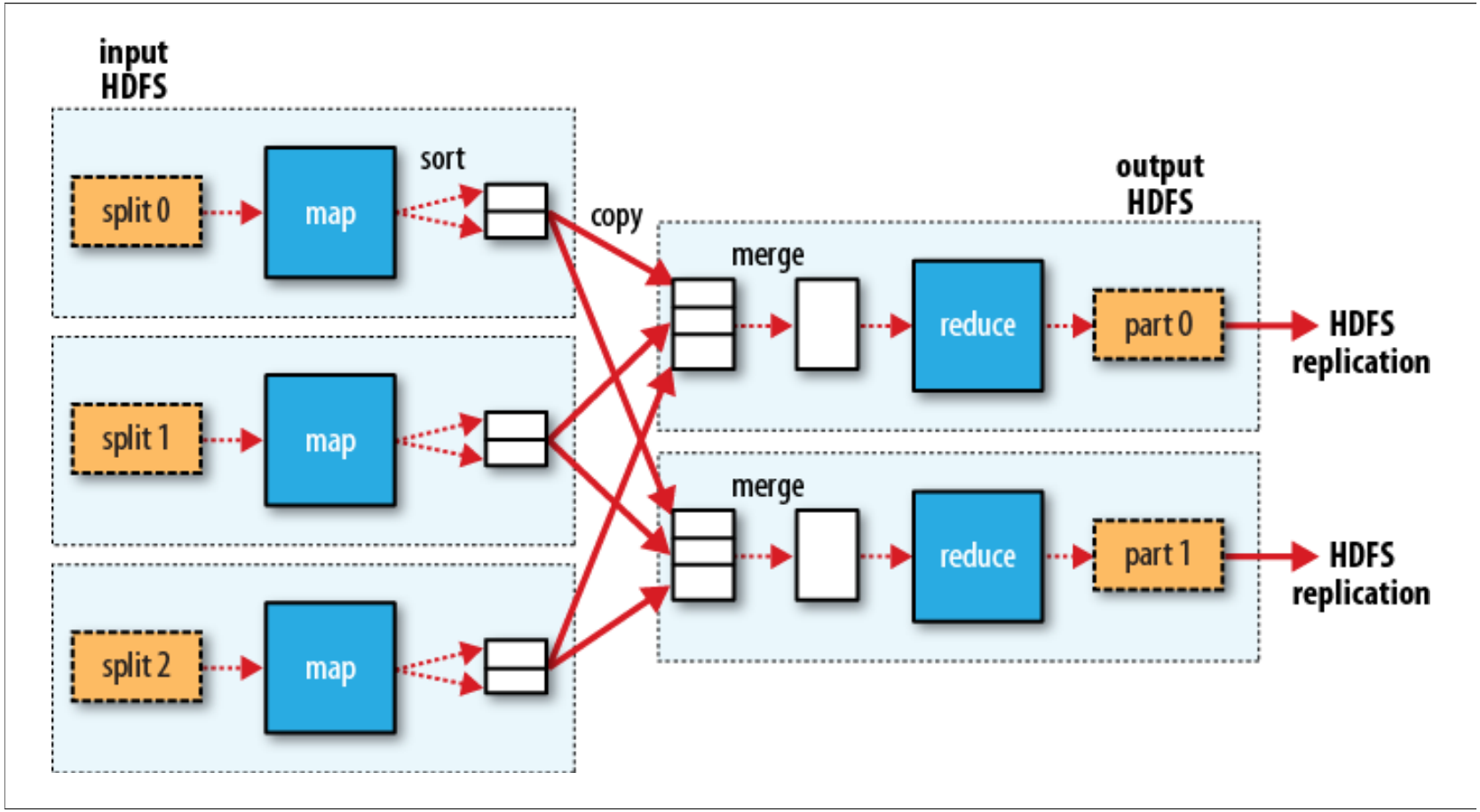
Partitioning

- What are some of the ways we might partition our user sessions?

- How would we do this?

MapReduce in Hadoop

Figure 2.4, Hadoop - The Definitive Guide

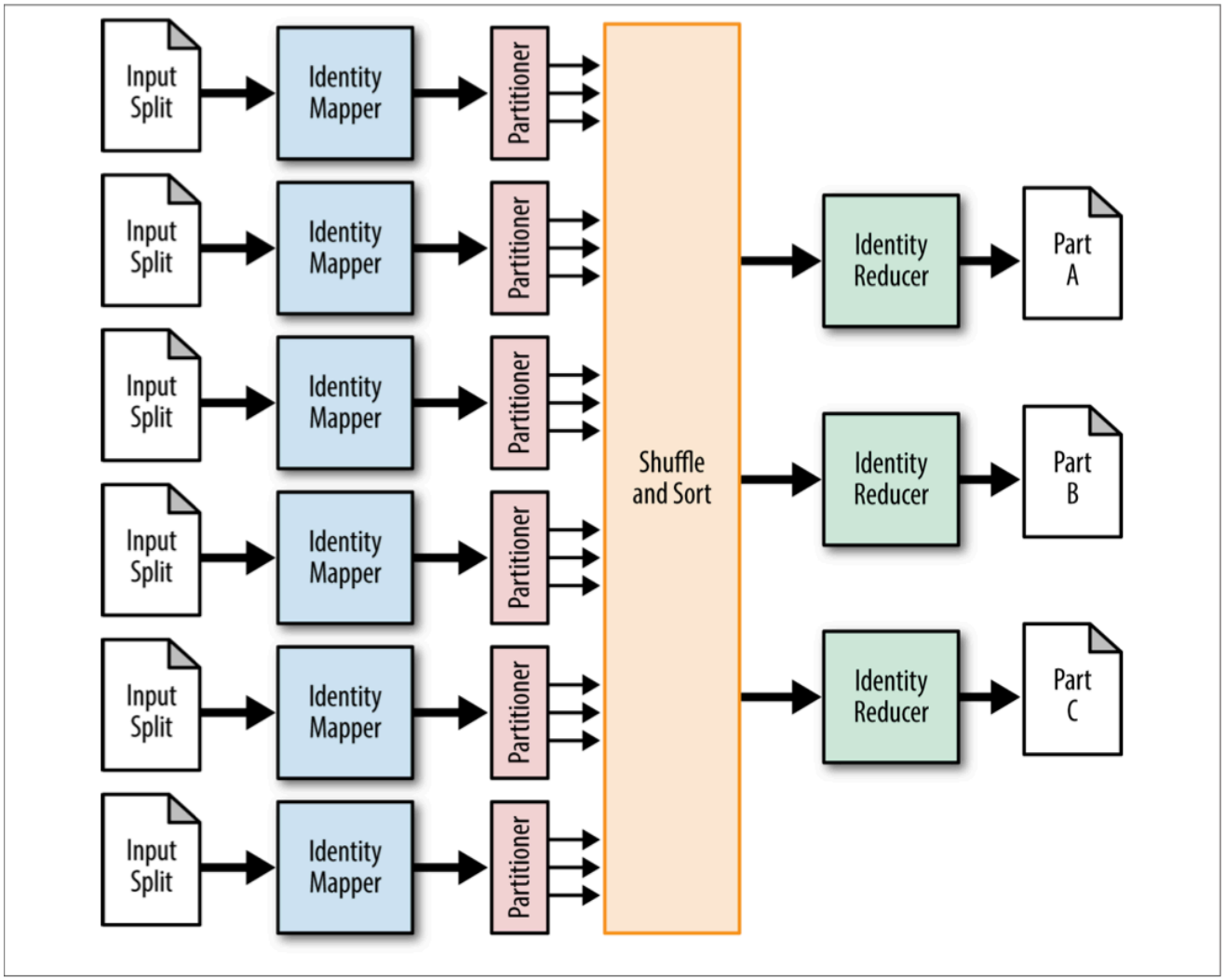


Partitioning

- Define a `Partitioner`
- Examines each `map()` output pair
- Computes a partition number

Data Flow

Figure 4-2 from MapReduce Design Patterns



Recommendations – Assign 6

- Run WordCount on dataSet6.txt – see what's in it
- Build your log entry parser and test it
  - Return a Map indexed by parameter name, with value being the parameter value
  - You can use this parser on other log types in the future
- Get you app working with just a few fields populated
  - Session with no impressions
  - Add impressions, but just the fields in the provided schema
  - Extend the schema and compile it
  - Then populate the new field(s) in your mapReduce code
- Write some unit tests as you go