Big data: architectures and data analytics

MapReduce patterns

Summarization Patterns

3

Summarization patterns

- Are used to implement applications that produce top-level/summarized view of the data
  - Numerical summarizations (Statistics)
  - Inverted index
  - Counting with counters

4

Summarization Patterns

Numerical Summarizations

5

Numerical Summarizations

- Goal
  - Group records/objects by a key field and calculate a numerical aggregate (average, max, min, standard deviation,..) per group
- Provide a top-level view of large input data sets
- Motivation
  - Few high-level statistics can be analyzed by domain experts to identify trends, anomalies, …
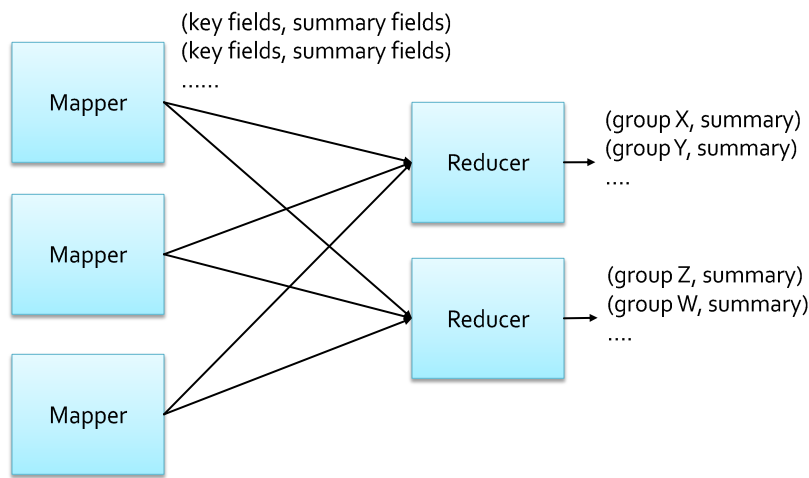
6

Numerical Summarizations - structure

- Mappers
  - Output (key, value) pairs where
    - key is associated with the fields used to define groups
    - value is associated with the fields used to compute the aggregate statistics
- Reducers
  - Receive a set of numerical values for each "group-by" key and compute the final statistics for each "group"
- Combiners
  - If the computed statistic has specific properties (e.g., it is associative), combiners can be used to speed up performances

7

Numerical Summarizations - structure

(key fields, summary fields)
(key fields, summary fields)
......

Mapper

Mapper

Mapper

Reducer

(group X, summary)
(group Y, summary)
....

Reducer

(group Z, summary)
(group W, summary)
....

8

Numerical Summarizations

- Known uses
  - Word count
  - Record count
  - Min/Max/Count
  - Average/Median/Standard deviation

9

Summarization Patterns

Inverted Index Summarizations

10

Inverted Index Summarizations

- Goal
  - Build an index from the input data to support faster searches or data enrichment
- Map terms to a list of identifiers
- Motivation
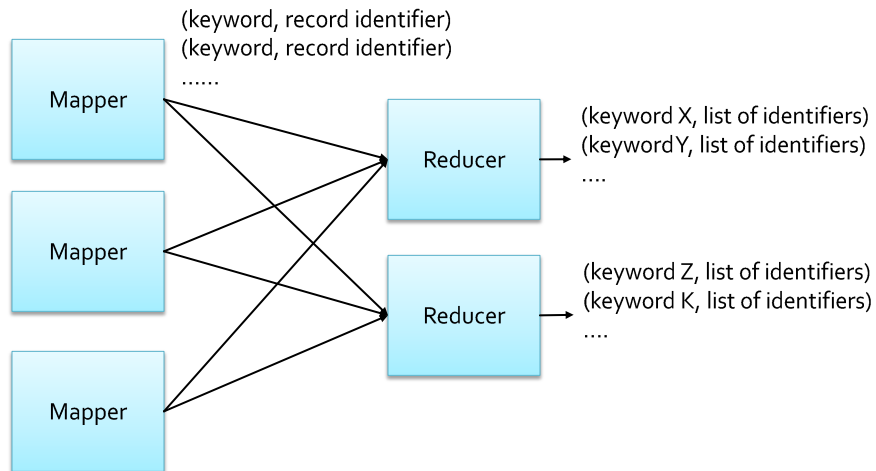 - Improve search efficiency

11

Inverted Index Summarizations - structure

- Mappers
  - Output (key, value) pairs where
    - key is the set of fields to index (a keyword)
    - value is a unique identifier of the objects to associate with each "keyword"
- Reducers
  - Receive a set of identifiers for each keyword and simply concatenate them
- Combiners
  - Usually are not useful when using this pattern
  - There is no values to aggregate

12

Inverted Index Summarizations - structure

```
Mapper

Mapper

Mapper
```

(keyword, record identifier)
(keyword, record identifier)
......

```
Reducer
```

(keyword X, list of identifiers)
(keyword Y, list of identifiers)
....

```
Reducer
```

(keyword Z, list of identifiers)
(keyword K, list of identifiers)
....

13

Inverted Index Summarizations

- Most famous known use
  - Web search engine
    - Word – List of URLs (Inverted Index)

14

Summarization Patterns

Counting with Counters

15

Counting with Counters

- Goal
  - Compute count summarizations of data sets
- Provide a top-level view of large data sets
- Motivation
  - Few high-level statistics can be analyzed by domain experts to identify trends, anomalies, …
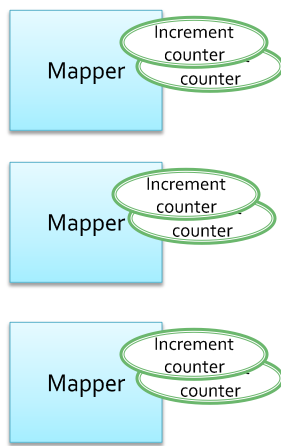
16

Counting with Counters - structure

- Mappers
  - Process each input record and increment a set of counters
- Map-only job
  - No reducers
  - No combiners
- The results are stored/printed by the Driver of the application

17

Counting with Counters - structure

18

9

Counting with Counters

- Known uses
  - Count number of records
  - Count a small number of unique instances
  - Summarizations

19

Filtering Patterns

20

Filtering Patterns

- Are used to select the subset of input records of interest
  - Filtering
  - Top K
  - Distinct

21

Filtering Patterns

Filtering

22

Filtering

- Goal
  - Filter out input records that are not of interest/keep only the ones that are of interest
- Focus the analysis of the records of interest
- Motivation
  - Depending on the goals of your application, frequently only a small subset of the input data is of interest for further analyses
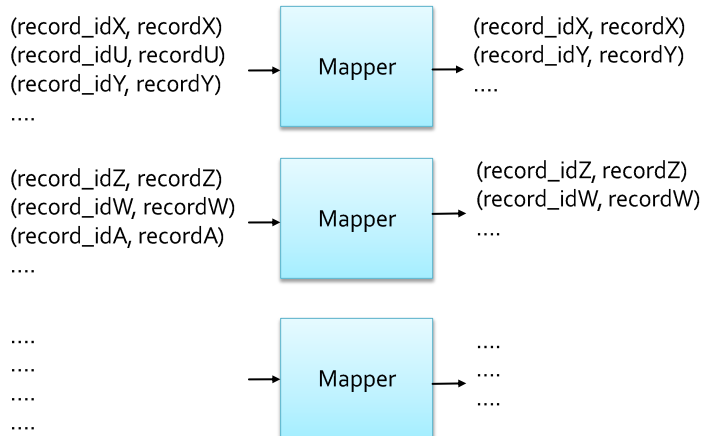
23

Filtering - structure

- The input of the mapper is a set of records
  - Key = primary key
  - Value = record
- Mappers
  - Output one (key, value) pair for each record that satisfies the enforced filtering rule
    - Key is associated with the primary key of the record
    - Value is associated with the selected record
  - Reducers
    - The reducer is useless in this pattern
    - A map-only job is executed (number of reduce set to 0)

24

Filtering - structure

(record_idX, recordX)
(record_idU, recordU)
(record_idY, recordY)
....

→ Mapper →

(record_idX, recordX)
(record_idY, recordY)
....

(record_idZ, recordZ)
(record_idW, recordW)
(record_idA, recordA)
....

→ Mapper →

(record_idZ, recordZ)
(record_idW, recordW)
....

....
....
....
....

→ Mapper →

....
....
....

25

Filtering

- Known uses
  - Record filtering
  - Tracking events
  - Distributed grep
  - Data cleaning

26

Filtering Patterns

Top K

27

Top K

- Goal
  - Select a small set of top K records according to a ranking function
- Focus on the most important records of the input data set
- Motivation
  - Frequently the interesting records are those ranking first according to a ranking function
    - Most profitable items
    - Outliers

28

Top K - structure

- Mappers
  - Each mapper initializes an in-mapper top k list
    - k is usually small (e.g., 10)
    - The current top k-records of each mapper can be stored in main memory
    - Initialization performed in the setup method of the mapper
  - The map function updates the current in-mapper top k list

29

Top K - structure

- Mappers
  - The cleanup method emits the k (key, value) pairs associated with the in-mapper local top k records
    - Key is the "null key"
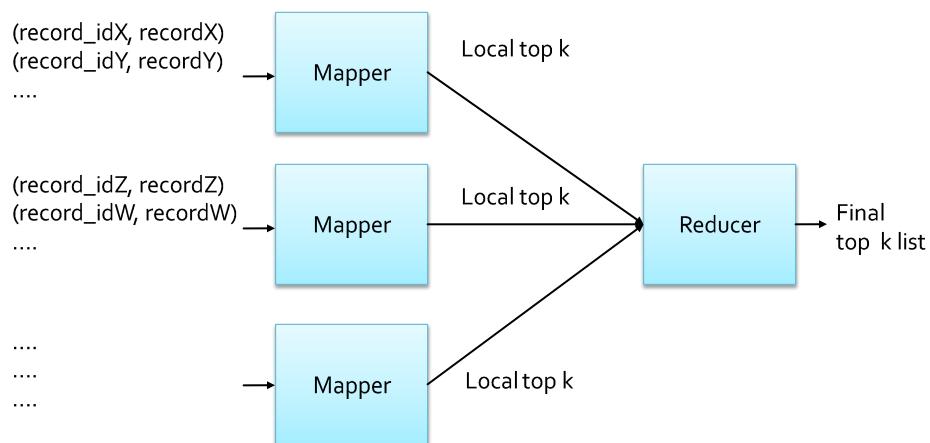   - Value is a in-mapper top k record

30

15

Top K - structure

- Reducer
  - A single reducer is instantiated
  - It computes the final top k list by merging the local lists emitted by the mappers
    - All input (key, value) pairs have the same key
    - Hence, the reduce method is called only once

31

Top K - structure

(record_idX, recordX)
(record_idY, recordY)
....
→ Mapper → Local top k

(record_idZ, recordZ)
(record_idW, recordW)
....
→ Mapper → Local top k

....
....
....
→ Mapper → Local top k

→ Reducer → Final top k list

32

Top K

- Known uses
  - Outlier analysis (based on a ranking function)
  - Select interesting data (based on a ranking function)

33

Filtering Patterns

Distinct

34

Distinct

- Goal
  - Find a unique set of values/records
- In some applications duplicate records are useless
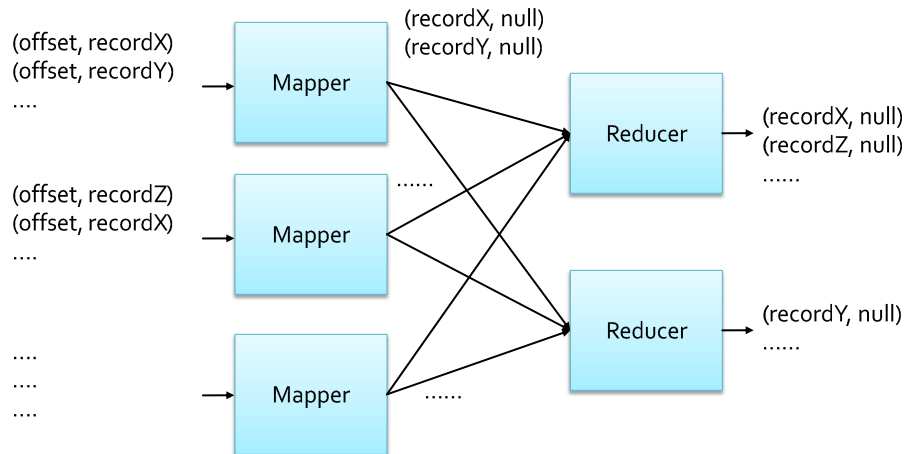- Motivation
  - Duplicates records are frequently useless

35

Distinct - structure

- Mappers
  - Emit one (key, value) pair for each input record
    - Key = input record
    - Value = null value
- Reducers
  - Emit one (key, value) pair for each input (key, list of values) pair
    - Key = input key, i.e., input record
    - Value = null value

36

Distinct - structure

(offset, recordX)
(offset, recordY)
....

Mapper

(recordX, null)
(recordY, null)

(offset, recordZ)
(offset, recordX)
....

Mapper

......

Reducer

(recordX, null)
(recordZ, null)
......

....
....
....

Mapper

......

Reducer

(recordY, null)
......

37

Distinct

- Known uses
  - Duplicate data removal
  - Distinct value selection

38