

Map Reduce and Design Patterns

Lecture 4

Fang Yu

Software Security Lab.
Department of Management Information Systems
College of Commerce, National Chengchi University
<http://soslab.nccu.edu.tw>

Cloud Computation, March 31, 2015



Data Organization Patterns

All about reorganizing data: Data will typically have to be transformed in order to interface nicely with the other systems. When migrating data from an RDBMS to a Hadoop system, one of the first things you should consider doing is reformatting your data into a more conducive structure.

- The structured to hierarchical pattern
- The partitioning and binning patterns
- The total order sorting and shuffling patterns
- The generating data pattern



Structured to Hierarchical

Transform your row-based data to a hierarchical format, such as JSON or XML

- `MutipleInputs` allows you to specify different input paths and different mapper classes for each input.
- The mappers load the data and parse the records into one cohesive format
- The reducer receives the data from all the different sources key by key. Build the hierarchical data structure from the list of data items. E.g., with XML or JSON, you'll build a single object and then write it out as output.
- Heap blow-out: all of those comments at one point might be stored in memory before writing the object out.



Structured to Hierarchical

Problem: Given a list of posts and comments, create a structured XML hierarchy to nest comments with their related post.

- We output the input value prepended with a character (P for a post or C for a comment)
- All the values are iterated to get the post record and collect a list of comments.x



Partitioning

The partitioning pattern moves the records into categories (i.e., shards, partitions, or bins) but it doesn't really care about the order of records.

- Partitioning means breaking a large set of data into smaller subsets, which can be chosen by some criterion relevant to your analysis.
- For example, in a HTTP server logs, you'll have GET and POST requests, internal system messages, and error messages. Analysis may care about only one category of this data
- Idea: Define the function that determines what partition a record is going to go to in a custom partitioner
- The custom partitioner will determine which reducer to send each record to; each reducer corresponds to particular partitions



Partitioning

Problem: Given a set of user information, partition the records based on the year of last access date, one partition per year.

- Configure: Use the custom built partitioner, e.g., 2008-2011, 4 reducers
- Mapper: $\langle \text{year}, \text{record} \rangle$. Set the category as the key and the record as the value
- Partition: Determine the partitions. The partitioner examines each key/value pair output by the mapper to determine which partition the key/value pair will be written. Each numbered partition will be copied by its associated reduce task during the reduce phase.
- Reducer: output record



Binning

The binning pattern, much like the previous pattern, moves the records into categories irrespective of the order of records.

- Binning splits data up in the map phase instead of in the partitioner
- Each mapper outputs one small file per bin
- Mapper only: having if-else statements to check each of the tags of a post. If the post contains the tag, it is written to the bin
- Use `MultipleOutputs`. Be sure to clean up.



Total Order Sorting

Sort your data in parallel on a sort key.

- Total order: If you concatenate the output files, the records are sorted
- Use a set of partitions divided by ranges of values
- Sort the data within a range



Total Order Sorting

Building the partition list via sampling and then performing the sort

- The analyze phase: To determine a set of partitions divided by ranges of values that will produce equal-sized subsets of data. Use random sampling on keys without values with one reducer
- The order phase: A custom partitioner is used to partition data by the sort key. The lowest range of data goes to the first reducer, the next range goes to the second reducer, so on and so forth. Use `TotalOrderPartitioner`
- Cost: load and parse the data twice



Shuffling

You have a set of records that you want to completely randomize.

- The mapper outputs the record as the value along with a random key.
- The reducer sorts the random keys, further randomizing the data.

