CS 696 Intro to Big Data: Tools and Methods
Fall Semester, 2016
Doc 23 Sorting & Partitioning
Nov 17, 2016

Issues

Structured data, binary format

Sorting data

- What if want to WordCount output sorted by count

Key-Value pairs

- One key & one value
- What if each record has N different items

Multiple Reducers

- How to balance load
- How to group data when has N different items
- How to global sort data so can
  - just append output files to get sorted data

Resources

Hadoop MapReduce v2 Cookbook 2nd Ed

Chapter 4 Developing Complex Hadoop MapReduce Applications
Through Secondary Sorting

Resources

http://blog.ditullio.fr/2015/12/18/hadoop-basics-working-with-sequence-files/

https://goo.gl/kE4yV1

Hadoop Basics

- Working with Sequence Files
- Filter, Aggregate, & Sort with MapReduce
- Secondary Sort
- Total Order Sorting in MapReduce
- Repartition Join in MapReduce
- Replicated join in MapReduce
- Bloom Filters
- Running SQL Queries with Hive

Example - Dataset

DonorsChoose project & donation database

1.6 GB csv file

_donationid
_projectid
_donor_acctid
_cartid
donor_city
donor_state
donor_zip
is_teacher_acct
donation_timestamp
donation_to_project
donation_optional_support
donation_total
dollar_amount
donation_included_optional_support
payment_method
payment_included_acct_credit
payment_included_campaign_gift_card
payment_included_web_purchased_gift_card
payment_was_promo_matched
via_giving_page for_honoree
thank_you_packet_mailed
donation_message

Sequence Files

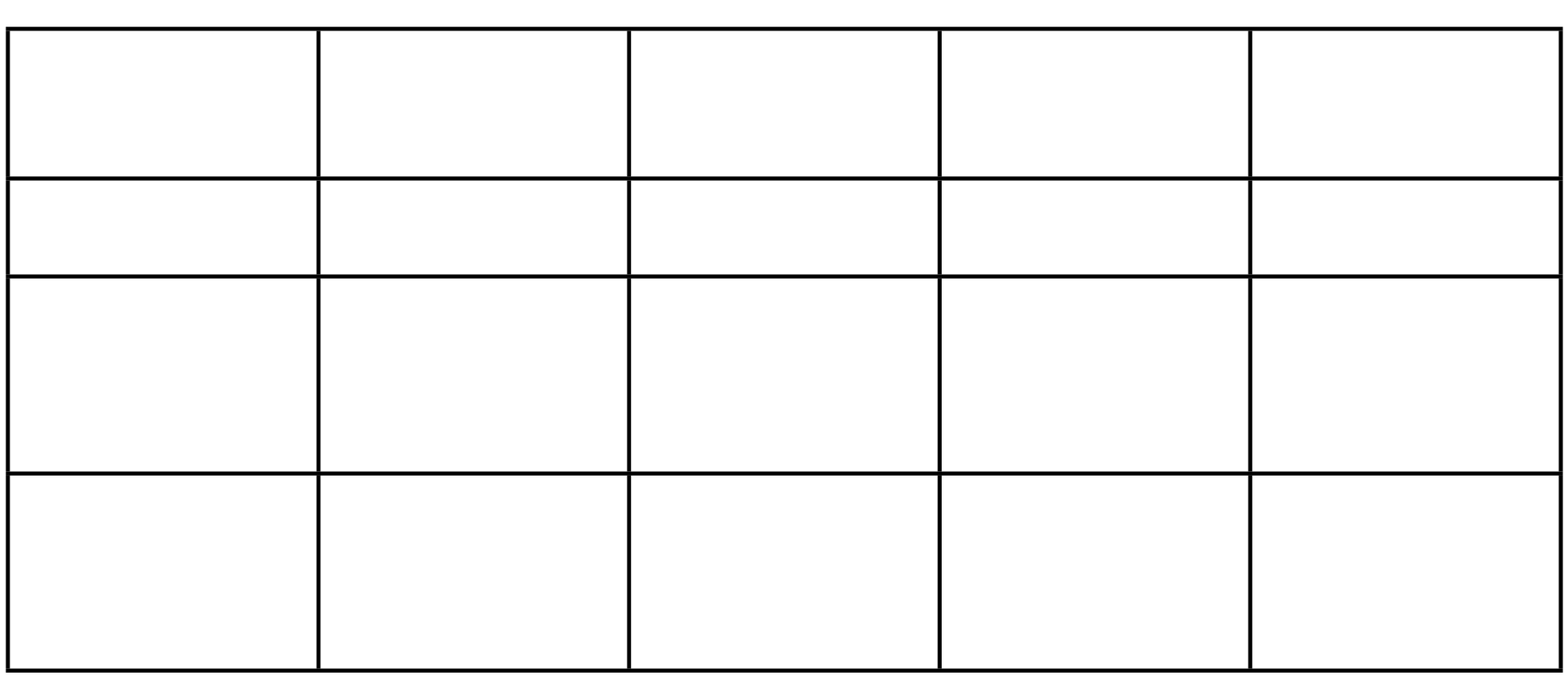
Why important - Shows how to deal with structured data

Sequence File

Sequence of binary key-value records

Don’t have to parse in map function

Supports compression for free



Filter, Aggregate and Sort

“View all donor cities by descending order of donation total amount, considering only donations which were not issued by a teacher. City names should be case insensitive (using upper-case)”

```
SELECT SUM(total) as sumtotal, UPPER(donor_city) as city
FROM donations
WHERE donor_is_teacher != 't'
GROUP BY UPPER(donor_city)
ORDER BY sumtotal DESC;
```

Filtering on the value of donor_is_teacher

Aggregating the sum of total values grouping by city

Sorting on the aggregated value sumtotal

First Job : Filtering and Aggregation

Map

Input : DonationWritables “full row” objects from the SequenceFile.

Output : (city, total) pairs if donor_is_teacher is not true.

Reduce

Reduce by summing the “total” values for each “city” key.

Second Job : Sorting

Map

Input : (city, sumtotal) pairs with summed total per city.

Output : (sumtotal, city) inversed pair.

Reduce

Identity reducer. Does not reduce anything,

but the shuffling will sort on keys for us.

Issue

Hadoop sorting uses increasing order

We want decreasing order

```
public static class DescendingFloatComparator extends WritableComparator {

    public DescendingFloatComparator() {
        super(FloatWritable.class, true);
    }

    @SuppressWarnings("rawtypes")
    @Override
    public int compare(WritableComparable w1, WritableComparable w2) {
        FloatWritable key1 = (FloatWritable) w1;
        FloatWritable key2 = (FloatWritable) w2;
        return -1 * key1.compareTo(key2);
    }
}
```

```
job.setSortComparatorClass(DescendingFloatComparator.class)
```

Secondary Sort

View the id, donor's state, donor's city and total donation amount for all donations which have a defined state and city of origin. Order the results by priority of :

State – ascending alphabetical order (case insensitive)

City – ascending alphabetical order (case insensitive)

Total amount – descending numerical order

```
SELECT donation_id, donor_state, donor_city, total
FROM donations
WHERE donor_state IS NOT NULL AND donor_city IS NOT NULL
ORDER BY lower(donor_state) ASC, lower(donor_city) ASC, total DESC;
```

Issue

Only have key & value - How to encode four items

Use tuple for key and/or value

Sorting is done on keys so make key

(state,city,total)

Implementing a custom Hadoop key type in Hadoop MapReduce Cookbook

How to sort the keys?

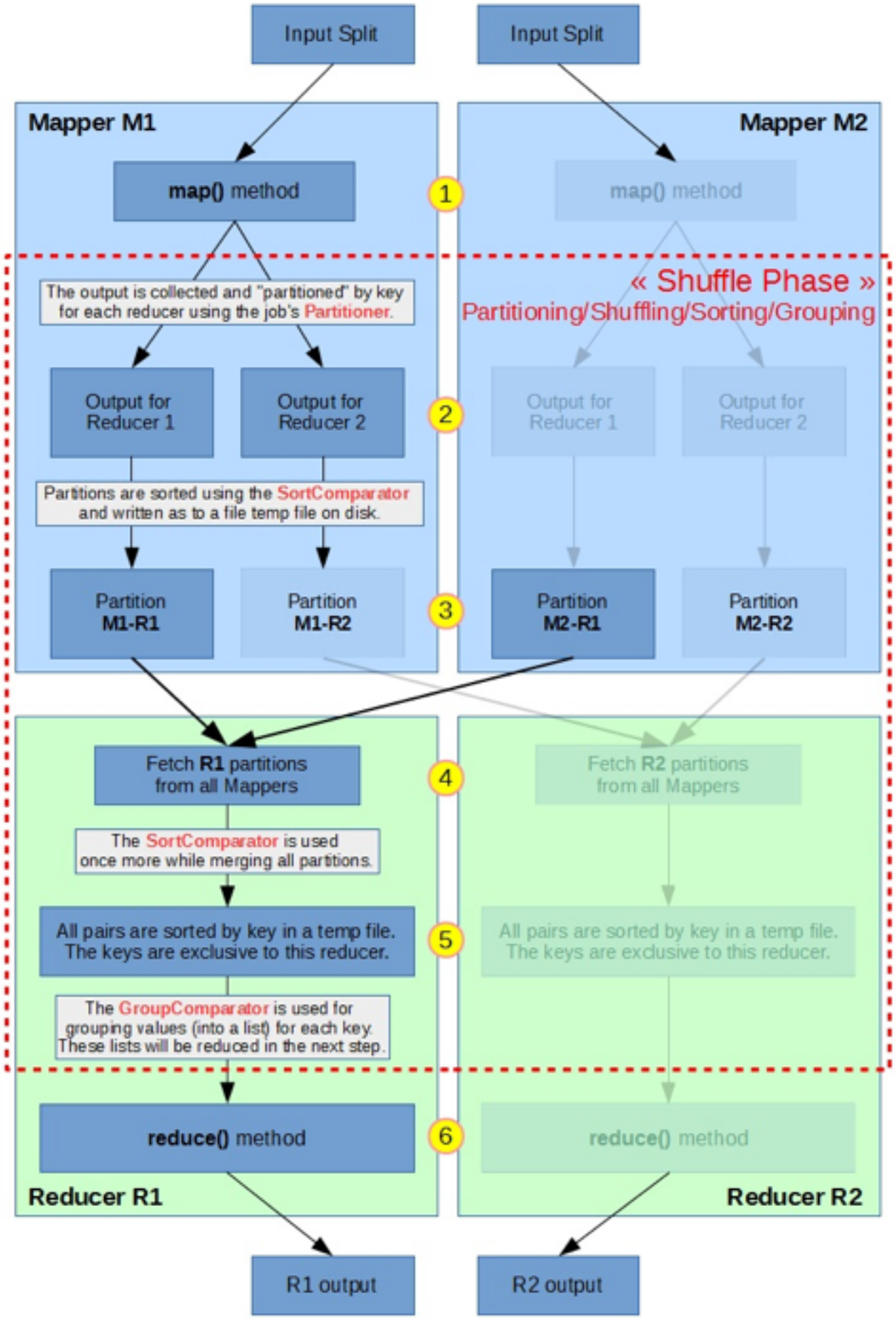
If have multiple reducers how to make sure

all keys with same state goes to same reducer?

How should reducer group the key-value pairs?

```
public class CompositeKey implements WritableComparable<CompositeKey> {

    public String state;
    public String city;
    public float total;

    public CompositeKey() { }

    public CompositeKey(String state, String city, float total) {
        super();
        this.set(state, city, total);  }

    public void set(String state, String city, float total) {
        this.state = (state == null) ? "" : state;
        this.city = (city == null) ? "" : city;
        this.total = total; }
```

```
public void write(DataOutput out) throws IOException {
        out.writeUTF(state);
        out.writeUTF(city);
        out.writeFloat(total); }

public void readFields(DataInput in) throws IOException {
        state = in.readUTF();
        city = in.readUTF();
        total = in.readFloat(); }

public int compareTo(CompositeKey o) {
        int stateCmp = state.toLowerCase().compareTo(o.state.toLowerCase());
        if (stateCmp != 0) {
                return stateCmp;
        } else {
                int cityCmp = city.toLowerCase().compareTo(o.city.toLowerCase());
                if (cityCmp != 0) {
                        return cityCmp;
                } else {
                        return Float.compare(total, o.total);
                }
        }
```



Partitioner

- Divides map output for reducers
- Default use keys hasCode()

SortComparator

GroupComparator

Partitioner

Want partitioner that sends data from a state to same reducer

```
import org.apache.hadoop.mapreduce.Partitioner;
import data.writable.DonationWritable;

public class NaturalKeyPartitioner extends Partitioner<CompositeKey, DonationWritable> {

    @Override
    public int getPartition(CompositeKey key, DonationWritable value, int numPartitions) {
        return Math.abs(key.state.hashCode() & Integer.MAX_VALUE) % numPartitions;
    }
}
```

```
job.setPartitionerClass(NaturalKeyPartitioner.class)
```

Issues with Partitioner

Data may not be spread evenly among reducers

If we what data sorted by state we will have to

- Merge output files
- Sort

Grouping

GroupComparator

```
import org.apache.hadoop.io.WritableComparable;
import org.apache.hadoop.io.WritableComparator;

public class NaturalKeyComparator extends WritableComparator {

    public NaturalKeyComparator() {
        super(CompositeKey.class, true);}

    public int compare(WritableComparable wc1, WritableComparable wc2) {
        CompositeKey key1 = (CompositeKey) wc1;
        CompositeKey key2 = (CompositeKey) wc2;
        return key1.state.compareTo(key2.state);
    }
}

job.setGroupingComparatorClass(NaturalKeyComparator.class)
```

Total Order Sorting in MapReduce

Manual Partitioning

TotalOrderPartitioner - partition on simple key types

Total Secondary Sorting

Manual Partitioning

Reducer 0 : state names starting with A to I (includes 9 letters)
Reducer 1 : state names starting with J to Q (includes 8 letters)
Reducer 2 : state names starting with R to Z (includes 9 letters)

```
import org.apache.hadoop.mapreduce.Partitioner;
import data.writable.DonationWritable;

public class CustomPartitioner extends Partitioner<CompositeKey, DonationWritable> {
    public int getPartition(CompositeKey key, DonationWritable value, int numPartitions) {
        if (key.state.compareTo("J") < 0) {
            return 0;
        } else if (key.state.compareTo("R") < 0) {
            return 1;
        } else {
            return 2;
        }
    }
}
```

TotalOrderPartitioner

Dynamically determines how to partition to balance load

InputSampler

- Samples data across all input splits
- Uses job's SortComparator to sort data
- Creates partition file to indicate how to partition data

TotalOrderPartitioner use partition file to send data to reducers

Types of InputSamplers

- RandomSampler
- IntervalSampler
- SplitSampler