BIG DATA ANALYTICS FRAMEWORK

UNIT II MAPREDUCE-I AND HDFS

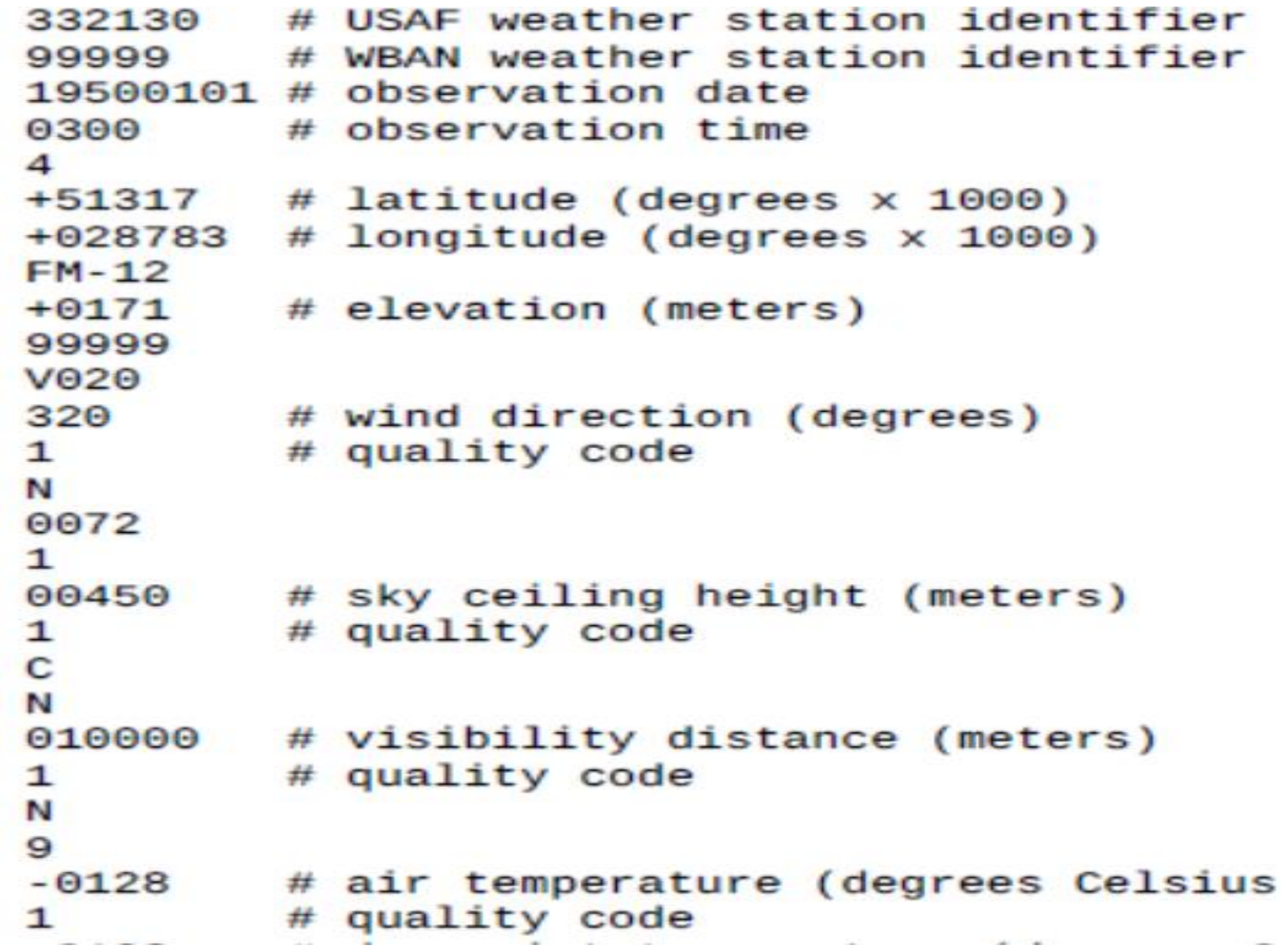
Apache Hadoop

- Open source software framework used to develop data processing applications which are executed in a distributed computing environment.
- Applications built using HADOOP are run on large data sets distributed across clusters of commodity computers.
- Commodity computers are cheap and widely available. These are mainly useful for achieving greater computational power at low cost.
- The processing model is based on 'Data Locality' concept wherein computational logic is sent to cluster nodes(server) containing data.
- This computational logic is nothing, but a compiled version of a program written in a high-level language such as Java. Such a program, processes data stored in Hadoop HDFS.

Analyzing Weather Dataset with Hadoop

- Weather sensors collecting data every hour at many locations across the globe gather a large volume of logdata, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented.
- The data we will use is from the National Climatic Data Center (NCDC, http://www .ncdc.noaa.gov/).
- The data is stored using a line-oriented ASCII format, in which each line is a record.

Format of a National Climatic Data Center record



```
332130   # USAF weather station identifier
99999    # WBAN weather station identifier
19500101 # observation date
0300     # observation time
4
+51317   # latitude (degrees x 1000)
+028783  # longitude (degrees x 1000)
FM-12
+0171    # elevation (meters)
99999
V020
320      # wind direction (degrees)
1        # quality code
N
0072
1
00450    # sky ceiling height (meters)
1        # quality code
C
N
010000   # visibility distance (meters)
1        # quality code
N
9
-0128    # air temperature (degrees Celsius
1        # quality code
```

Sample lines of input data

- 0067011990999991950051507004...9999999N9+00001+99999999999...
- 0043011990999991950051512004...9999999N9+00221+99999999999...
- 0043011990999991950051518004...9999999N9-00111+99999999999...
- 0043012650999991949032412004...0500001N9+01111+99999999999...
- 0043012650999991949032418004...0500001N9+00781+99999999999...
- These lines are presented to the map function as the key-value pairs:

(0, 006701199099999**1950**051507004...9999999N9+**0000**1+99999999999...)
(106, 004301199099999**1950**051512004...9999999N9+**0022**1+99999999999...)
(212, 004301199099999**1950**051518004...9999999N9**-0011**1+99999999999...)
(318, 004301265099999**1949**032412004...0500001N9**+0111**1+99999999999...)
(424, 004301265099999**1949**032418004...0500001N9**+0078**1+99999999999...)

The map function merely extracts the year and the air temperature (indicated in bold text), and emits them as its output (the temperature values have been interpreted asintegers):

- (1950, 0)
- (1950, 22)
- (1950, −11)
- (1949, 111)
- (1949, 78)

- The output from the map function is processed by the MapReduce framework before being sent to the reduce function. This processing sorts and groups the key-value pairs by key. So, continuing the example, our reduce function sees the following input:

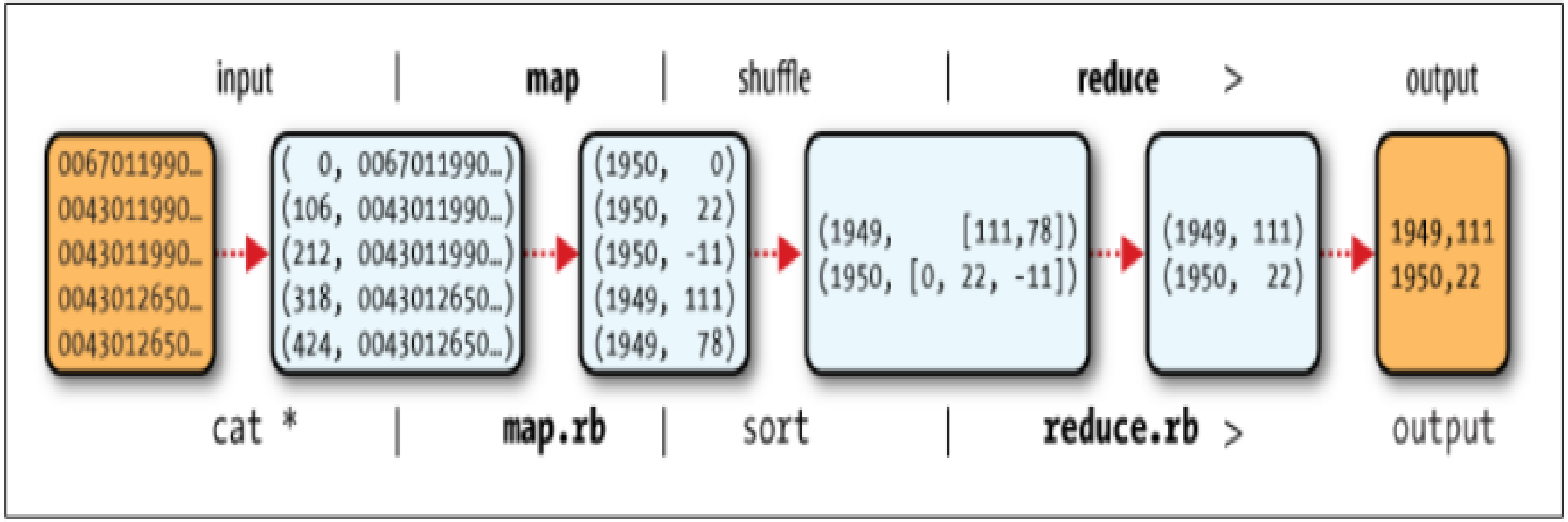
(1949, [111, 78])

(1950, [0, 22, −11])

- Each year appears with a list of all its air temperature readings. All the reduce function has to do now is iterate through the list and pick up the maximum reading:

(1949, 111) (1950, 22)

MapReduce logical data flow



Amazon Sales Report

- Total sales citywise for the year 2022 in India