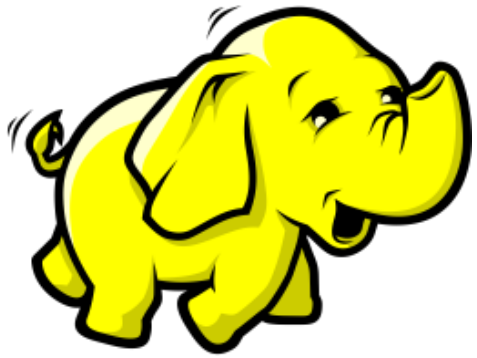
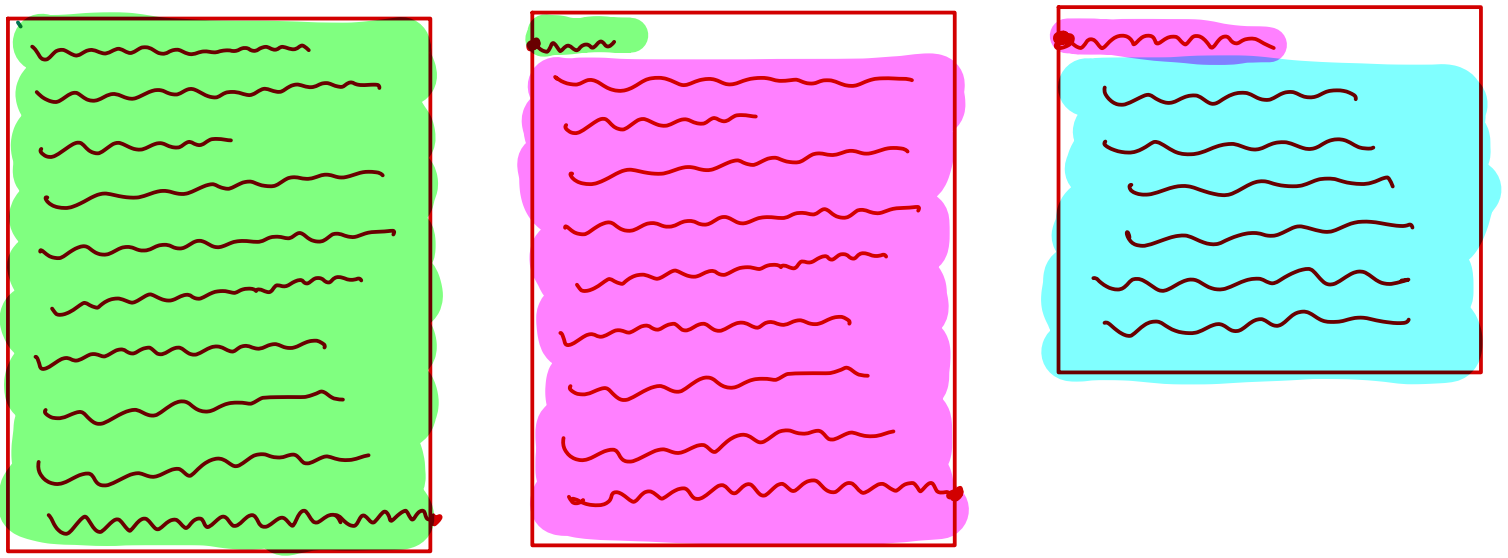


Big Data – Hadoop

Trainer: Mr. Nilesh Ghule.

emp.csv (300 mb)



input splits ≃ hdfs blocks.

Implementing MR job

- Implement Mapper class
  - Handle individual record
  - class MyMapper extends Mapper<KeyIn, ValueIn, KeyOut, ValueOut> { … }
  - Override map() method
  - Input from InputFormat record by record and key-value pair output to merge stage
- Implement Reducer class
  - Perform aggregation on set of values (corresponding to each key)
  - class MyReducer extends Reducer<KeyIn, ValueIn, KeyOut, ValueOut> { … }
  - Override reduce() method
  - Input from merge stage in key-values pair and key-value pair output to OutputFormat
- Hadoop Writable
  - Like java wrapper classes, but optimized for serialization over the network.
  - IntWritable, ByteWritable, ShortWritable, LongWritable, DoubleWritable, BooleanWritable, Text
  - ArrayWritable, MapWritable, NullWritable

Implementing MR job

- Create MR job
  - Job and Jar
  - Mapper class & its output
  - Reducer class & its output
  - Input & Output format
  - Combiner, Partitioner
  - Submit job
- Configured class
  - Associate configuration object with the driver
  - getConf() and setConf()
- Tool and ToolRunner
  - Tool is standard way of implementing any processing on Hadoop – run() method
  - ToolRunner is helper to execute the Tool.
- Generic options
  - hadoop jar <jar-path> <generic-options> <cmd-line args to main-class>
  - hadoop jar <jar-path> <main-class> <generic-options> <cmd-line args to main-class>
  - Generic options:
    - -conf
    - -D
    - -fs
    - -jt
    - -files
    - -libjars
  - GenericOptionsParser

Executing MR

- hadoop jar command
  - hadoop jar <jar-path> <generic-options> <cmd-line args to main-class>
  - hadoop jar <jar-path> <main-class> <generic-options> <cmd-line args to main-class>
- MR job configurations
  - fs.defaultFS = hdfs://namenode:9000/
  - mapreduce.framework.name = yarn
  - yarn.resourcemanager.address = resourcemanager:8032
- Understanding MR summary
  - Number of mapper & reducer tasks.
  - Number of input & output records for mapper
  - Number of input & output records for reducer
  - Custom Job counters
- MR log files review
  - $HADOOP_HOME/logs

Input/Output Format and Input Splits

- InputFormat – how to read data
  - FileInputFormat, TextInputFormat
  - KeyValueTextInputFormat, NLineInputFormat
  - DBInputFormat
- RecordReader – Logical division of record
- Number of mappers = Number of input splits
- Number of input splits ≈ Number of HDFS blocks
- Output format – how to write data
  - FileOutputFormat, TextOutputFormat
  - DBOutputFormat
- RecordWriter – Write individual record
- Number of output files = Number of reducers
- Output written on HDFS (replicated)

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>