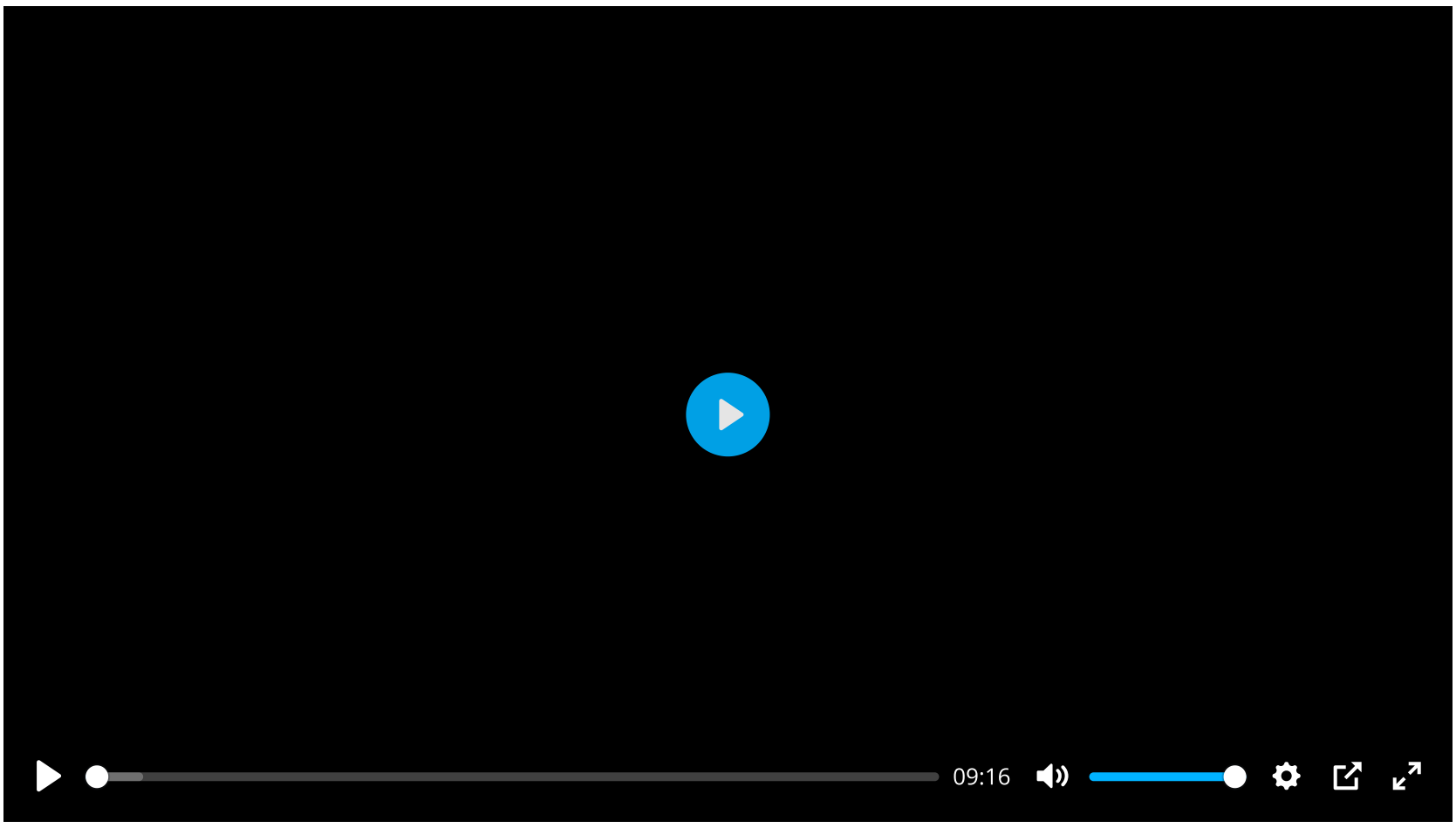
3.3 Технология Map-Reduce



Текстовая расшифровка видео:

СЕКРЕТНЫЕ ОПТИМИЗАЦИИ

План:

- Combiner, Partitioner, Comparator, Compression;
- Количество мапперов и редюсеров.

Combiner, Partitioner, Comparator, Compression

Combiner – промежуточный код, который запускается между маппером и редюсером. Может предагрегировать данные перед пересылкой их на редюсер.

Comparator – возможность указать более точно механизм сравнения ключей при шаффле и сортировке.

Пример того, как запускать job'у с comparator:

```
~$ yarn jar $HADOOP_STREAMING_JAR \
          -D stream.num.map.output.key.fields=2 \
          -D stream.num.reduce.output.key.fields=2 \
          -D mapreduce.job.output.key.comparator.class=org.apache.hadoop.mapreduce.lib.parti
          -D mapreduce.partition.keycomparator.options="-k1,1n -k2,2" \
```

Наши данные могут быть нетривиальной структуры, и мы можем захотеть их отсортировать, например, по нестандартному ключу и т.д. Поэтому, когда мы используем hadoop streaming, для этих целей напрямую написать comparator на Python мы не сможем, но мы сможем указать класс Java и аргументы, которые будем использовать в comparator.

В примере в параметрах указан ключ и у маппера, и после маппера, и на входе редюсера может иметь два поля. Далее, для сравнения значений между собой использован класс Java – KeyFieldBasedComporator как для ключей, составных из двух филдов.

Partitioner – дополнительная логика, которая позволяет подтюнить партишенинг, например, в случае сложного составного ключа.

```
~$ yarn jar $HADOOP_STREAMING_JAR \
            -D stream.num.map.output.key.fields=2 \
            -D stream.num.reduce.output.key.fields=2 \
            -D mapreduce.map.output.key.field.separator=.
            -D
            mapreduce.job.output.key.comparator.class=org.apache.hadoop.mapreduce.lib.par
            tition.KeyFieldBasedComparator \
            -D mapreduce.partition.keycomparator.options="-k1,1nr -k2,2" \
            -D mapreduce.partition.keypartitioner.options=-k2.1,2.1 \
            -partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner \
```

В случае с Partitioner мы можем задать следующее: используем сортировку по ключу через comparator (можно указать на separator, который тоже используется между филдами для разграничения) и говорим о Partitioner, используя **KeyFieldBasedPartitioner**.

Полезная плюшка: параметры «-k1, 1n» и «-k2, 2"» выглядят в случае использования KeyFieldBasedComporator точно так же, как ключи у команды «Sort».

Количество мапперов и редюсеров

Количество мапперов по умолчанию зависит **от разбиения входных данных**, а разбиение зависит от размера блока (вспомним про 128мб на HDFS). Тем не менее, можно указать количество тасков.

Если выставить **количество редюсеров** в 0 – получится **map-only**-задача, которая может быть нужна, например, для фильтрации.

```
… -D mapred.map.tasks=5 -D mapred.reduce.tasks=2 …
 # также работает -numReduceTasks 1
```