summary of section 2.4

Jixiang Hao

March 2020

1 workflow system

workflow can be saw as an acyclic graph representing workflow. For example, the MapReduce is a two-step workflow where the output of map function feed into reduce function, but if we use more than 2 functions, the the structure of workflow will be a acyclic graph: there are lot of big data systems implemented bash on the workflow system such as the Spark, Tensorlfow.

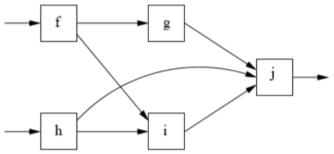

Figure 1: asclic graph

2 spark

2.1 introduction

spark is an advance workflow system, the publication of spark can solve the lot of problems existing in HDFS, the basic architecture of spark shows below: The combination of spark include Driver Program, Executor, and Tasks. The Work Nodes are not necessary because the Spark reduces the coupling with the file system, Spark can use the task-Manager provided by yarn as the file system.

2.2 Resilient Distributed Dataset(RDD)

RDD is the central data abstraction of spark, for understand what is RDD, the pictures below compare the java IO process with process of counting words count by using spark, we can find 2 type of process are similar and different. The transformation are used in both process, the file are transferred into different functions, but the java IO process return different type of object while the spark

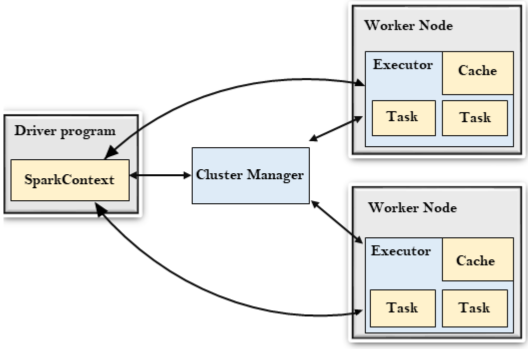

Figure 2: Spark Architecture

only return the RDD objects, and the RDD is generally, a dataset that need be computing based on task. Unlike the Hadoop MapRecuce abstract the data into key value pair, the spark do not restrict the type of data comprised in RDD, In other words, you can directly process the data in RDD, but you can not directly process data from the map part because the data has transferred into key value pairs.

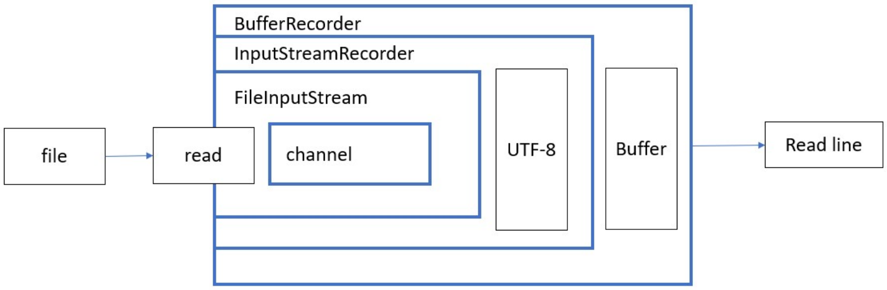

Figure 3: Java IO Process

2.3 Operation

the Operation is important in spark, which undertakes Spark's main data logic processing and data transmission. there are two kind of operations, transformation and action. for transformation, the book mentioned two transformations, Map and Flatmap, and two actions, Reduce and GroupByKey, the Map transformatiom takes a function as parameter, which operate data in a specific way. Map returns an RDD.The RDD is composed of each input element transformed

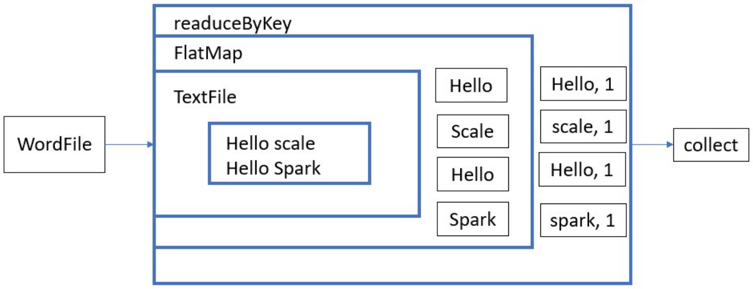

Figure 4: spark

by the parameter function. for example, the Scale code below show the a array adding operation for RDD:

```
\\ mapping source with adding 2
val source = makeRDD(1 to 10)
val mapped = source.map(_+2)
```

the code means that creating a new RDD with type array, and mapping every elements of RDD to add 2 and returning a RDD with processed data. In Spark, the map can apply to any type object, but it can exactly return one result which is the RDD object, if we want to return a set of objects from a single object, the flatmap transformation is a good choice, which similar to Map of MapReduce but return RDD of sequence type rather than key value pair.

```
\\ mapping each elements in RNN and returning a RDD with flat shape
val listRDD : RDD[List[Int]] = makeRDD(Array(List(1, 2), List(3, 4)))
val faltMapRDD: RDD[Int] = ListRDD.flatmap(datas=>datas)
```

filter is a restriction that filter the data of RDD from in input. For example, the sacale below show the filter transformation:

```
\\ filtering each elements in RNN and returning a new RDD object
val listRDD : RDD[List[Int]] = makeRDD(list(1, 2, 3, 4))
val FilterMapRDD: RDD[Int] = ListRDD.Filter(datas=>datas%2==0)
```

the code means that creating a new RDD wirh type list, and drop the elements that can not be divisible by 2.
the Reduce operation is an action, not a transformation, which return the value but not the RDD obejct. the Reduce operation can reduce the elements in RDD to a single element.

2.4 Spark implementation

there are many different ways to implement Spark, the book meanly talk talk two types, we meanly introduce the lazy evaluation of RDD. In the lazy evaluation, the RDDs do not accumulate in a data node until all RDDs completed process, the RDDs flow to other compute node that do same computing to apply other transformation. As a result, the the RDDs are not store in disk where the read write speed will improve because of low IO process.

3 Tensorflow

Like Spark, TensorFlow provides a programming interface where one writes a sequence of steps. Programs are typically acyclic, although like Spark the mean data in tensorflow called tensor, which transfer between blocks of code. One difference between Spark and TensorFlow is the different of data type. In place of the RDD, but tensors in Tensorflow.