Mini Unicorn Operators

Erman Yafay

April 7, 2016

1 Introduction

Document explains the execution stages of Mini Unicorn's operators with examples. Initial data is a distributed collection of type;

$$RDD[(Term, \ List[Hit])]$$

where RDD represents a distributed collection of $(Term, \ List[Hit])$ key-value pairs. Consequently, $Term$ and Hit are composed of the following data structures;

$$
\begin{aligned}
Term & \rightarrow & (edge - type, \ id) \\
Hit & \rightarrow & (DocId, \ hit - data) \\
DocId & \rightarrow & (vertex - type, \ id, \ rank)
\end{aligned}
$$

For simplicity, all the edge-type's and vertex-type's are omitted. Following data collection will be used for all the examples. It will be referred as **postingLists**.

$$
\begin{aligned}
\textbf{id} & \rightarrow & \textbf{List of Hits} \\
\{ \ 0 & \rightarrow & [\ (1, \ 0.35) \ ], \\
1 & \rightarrow & [\ (5, \ 0.54), \ (4, \ 0.26), \ (0, \ 0.16) \ ], \\
2 & \rightarrow & [\ (5, \ 0.54) \ ], \\
3 & \rightarrow & [\ (6, \ 0.35) \ ], \\
4 & \rightarrow & [\ (1, \ 0.35), \ (6, \ 0.35) \ ], \\
5 & \rightarrow & [\ (7, \ 0.37), \ (1, \ 0.35), \ (6, \ 0.35), \ (8, \ 0.33), \ (2, \ 0.22) \ ] \ \}
\end{aligned}
$$

where symbols denote;

$$
\begin{aligned}
\{\} &= RDD\ collection \\
\rightarrow &= key - value\ seperator \\
[] &= List\ collection \\
() &= id - rank\ pair
\end{aligned}
$$

2 Term Operator

2.1 Example

Friends of 4.

```
scGraph.term(Term(FriendEdge, 4))
```

2.2 Scala Implementation

SocialGraph.scala

```
def term(term: Term) = {
  postingLists.filter { case (t, lh) =>
  term == t }.flatMap(_._2).sortBy(_.docId.rank, false)
}
```

2.2.1 filter

```
filter { case (t, lh) => term == t }
```

Select the element whose id is 4.

$$
\begin{aligned}
Type &= RDD[(Term,\ List[Hit])] \\
Data &= \{\ 4 \rightarrow [\ (1,\ 0.35),\ (6,\ 0.35)\ ]\ \}
\end{aligned}
$$

2.2.2 flatMap

```
flatMap(_._2)
```

Retrieves the adjacency list of 4 as an RDD. Unfortunately, sorted order is not preserved.

$$
\begin{aligned}
Type &= RDD[Hit] \\
Data &= \{\ (1,\ 0.35),\ (6,\ 0.35)\ \}
\end{aligned}
$$

2.2.3 sortBy

```
sortBy(_.docId.rank, false)
```

Sorts the RDD in descending order by ranks.

$$
\begin{aligned}
Type &= RDD[Hit] \\
Data &= \{ (1,\ 0.35),\ (6,\ 0.35) \}
\end{aligned}
$$