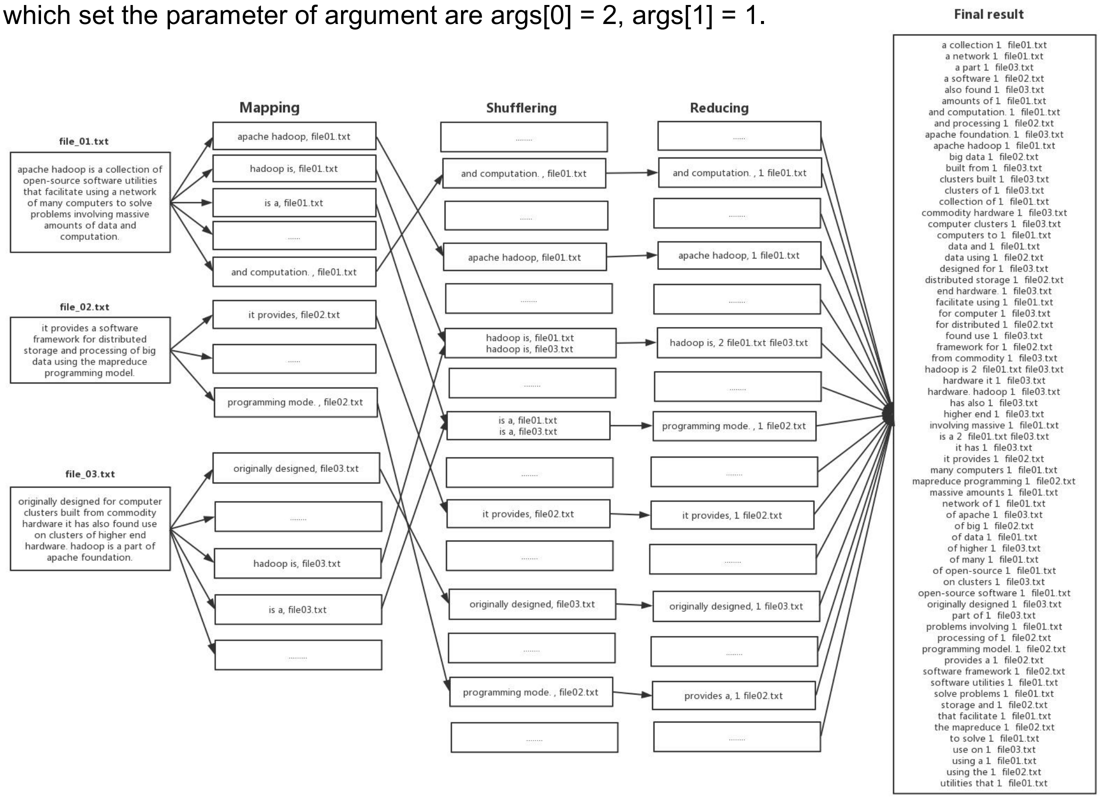
COMP9313 Assignment1 Report

1. Map and Reduce Process

As the graph show below, you can see the design process of map and reduce in assignment 1 which set the parameter of argument are args[0] = 2, args[1] = 1.


Mapping
Shufflering
Reducing
Final result
file_01.txt
apache hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation.
file_02.txt
it provides a software framework for distributed storage and processing of big data using the mapreduce programming model.
file_03.txt
originally designed for computer clusters built from commodity hardware it has also found use on clusters of higher end hardware. hadoop is a part of apache foundation.
apache hadoop, file01.txt
hadoop is, file01.txt
is a, file01.txt
......
and computation. , file01.txt
it provides, file02.txt
......
programming mode. , file02.txt
originally designed, file03.txt
....
hadoop is, file03.txt
is a, file03.txt
....
....
and computation. , file01.txt
......
apache hadoop, file01.txt
....
hadoop is, file01.txt
hadoop is, file03.txt
....
is a, file01.txt
is a, file03.txt
....
it provides, file02.txt
....
originally designed, file03.txt
....
programming mode. , file02.txt
....
......
and computation. , 1 file01.txt
....
apache hadoop, 1 file01.txt
....
hadoop is, 2 file01.txt file03.txt
....
programming mode. , 1 file02.txt
....
it provides, 1 file02.txt
....
originally designed, 1 file03.txt
....
provides a, 1 file02.txt
....

```
a collection 1  file01.txt
a network 1  file01.txt
a part 1  file03.txt
a software 1  file02.txt
also found 1  file03.txt
amounts of 1  file01.txt
and computation. 1  file01.txt
and processing 1  file02.txt
apache foundation. 1  file03.txt
apache hadoop 1  file01.txt
big data 1  file02.txt
built from 1  file03.txt
clusters built 1  file03.txt
clusters of 1  file03.txt
collection of 1  file01.txt
commodity hardware 1  file03.txt
computer clusters 1  file03.txt
computers to 1  file01.txt
data and 1  file01.txt
data using 1  file02.txt
designed for 1  file03.txt
distributed storage 1  file02.txt
end hardware. 1  file03.txt
facilitate using 1  file01.txt
for computer 1  file03.txt
for distributed 1  file02.txt
found use 1  file03.txt
framework for 1  file02.txt
from commodity 1  file03.txt
hadoop is 2  file01.txt file03.txt
hardware it 1  file03.txt
hardware. hadoop 1  file03.txt
has also 1  file03.txt
higher end 1  file03.txt
involving massive 1  file01.txt
is a 2  file01.txt file03.txt
it has 1  file03.txt
it provides 1  file02.txt
many computers 1  file01.txt
mapreduce programming 1  file02.txt
massive amounts 1  file01.txt
network of 1  file01.txt
of apache 1  file03.txt
of big 1  file02.txt
of data 1  file01.txt
of higher 1  file03.txt
of many 1  file01.txt
of open-source 1  file01.txt
on clusters 1  file03.txt
open-source software 1  file01.txt
originally designed 1  file03.txt
part of 1  file03.txt
problems involving 1  file01.txt
processing of 1  file02.txt
programming model. 1  file02.txt
provides a 1  file02.txt
software framework 1  file02.txt
software utilities 1  file01.txt
solve problems 1  file01.txt
storage and 1  file02.txt
that facilitate 1  file01.txt
the mapreduce 1  file02.txt
to solve 1  file01.txt
use on 1  file03.txt
using a 1  file01.txt
using the 1  file02.txt
utilities that 1  file01.txt
```

In Mapping function, I design the <Input key, Input value, output key, output value> is <Object, Text, Text, Text>. Firstly, Divide the contents of the file into words by space, and then use the iterator of the ngrams algorithm to sequentially obtain phrases consisting of n consecutive words. Next, write the key-value pairs(ngrams word, file name which word belong to) to context.

In Reducing function, The <Input key, Input value, output key, output value> is <Text, Text, Text, Text>. In the beginning, get the shufflering and sorting result <ngarms word, list of file name>. After that calculate the length of the list, we know the occurrences number of ngrams word. Next, I connect the length and the file name that appears in the file as the output value. The output key is ngrams word which is also input key. Finally, I write the output key-value pair into context.

2. Main(Driver)

In the main function, I creating a Configuration object and a Job object, assigning a job name for identification purposes. Then setting some attributes of job. After that, the HDFS input and output directory to be fetched from the command line, and two parameters for program. Finally, Submit the job to the cluster and wait for it to finish.