# Chameleon Cloud Tutorial

National Science Foundation

Program Solicitation # NSF 13-602

CISE Research Infrastructure: Mid-Scale Infrastructure - NSFCloud (CRI: NSFCloud)

# Hadoop - Map Reduce (Python)

In this tutorial, we will discuss about the Map and Reduce program, its implementation.

| # | Action | Detail | Time (min) |
|---|---|---|---|
| 1 | Implementation of multiple MapReduce programs | You will learn how to implement MapReduce programs on everyday scenario’s | 15 |
| 2 | Execution of MapReduce programs | In this session you will get familiar on how to execute the MapReduce program and the analysis of the input data. | 10 |

# Prerequisites

The following prerequisites are expected for successful completion of this tutorial:

- Chameleon Cloud account (http://chameleoncloud.org/user/register/)
- SSH client (Windows users: download PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html))
- A basic knowledge of Linux.
- Installation of Hadoop and Map reduce.
- Single node set up.

# Step 1: Implementation of MapReduce

Login to the system with user id ‘CC’ and enter to establish the connection.


Enter to establish connection (Authenticate with the public key)

```
login as: cc
Authenticating with public key "rsa-key-20150615"
Last login: Mon Jun 29 11:19:55 2015 from cpe-72-191-37-37.satx.res.rr.com
[cc@karthi-khj059-n01 ~]$
```

Change the directory to MapReduce-Basics-master (hadoop-mapreduce-MaReduce-Basics-master)

```
[cc@karthi-khj059-n01 ~]$ cd hadoop
[cc@karthi-khj059-n01 hadoop]$ cd mapreduce
[cc@karthi-khj059-n01 mapreduce]$ cd MapReduce-Basics-master
[cc@karthi-khj059-n01 MapReduce-Basics-master]$
```

View the files present in the MapReduce-Basics-master directory using ls command.

```
[cc@karthi-khj059-n01 ~]$ cd hadoop
[cc@karthi-khj059-n01 hadoop]$ cd mapreduce
[cc@karthi-khj059-n01 mapreduce]$ cd MapReduce-Basics-master
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ ls
assymetric_friendships.py  friends_count.py  MapReduce.py   README.md
data                       invert_index.py   MapReduce.pyc  solutions
dna.py                     join.py           multiply.py    wordcount.py
[cc@karthi-khj059-n01 MapReduce-Basics-master]$
```

Move to the data directory where the input data is stored.

```
[cc@karthi-khj059-n01 MapReduce-Basics-master]$
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ ls
assymetric_friendships.py  friends_count.py  MapReduce.py   README.md
data                       invert_index.py   MapReduce.pyc  solutions
dna.py                     join.py           multiply.py    wordcount.py
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ cd data
[cc@karthi-khj059-n01 data]$ ls
books.json  dna.json  friends.json  matrix.json  records.json
[cc@karthi-khj059-n01 data]$
```

# Inverted Index:

In this the input Json file contains document _id and the sample text, where

**document_id** - An document identifier formatted as string

**text** - Document formatted as a string.

Output file contains (word, document ID list) where word is an string and the document ID list is list of strings.

**Input text file:**

Open the data directory to view the input json file. Use the below command shown in the screen print to view the books.json file.

```
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ cd data
[cc@karthi-khj059-n01 data]$ ls
books.json  dna.json  friends.json  matrix.json  records.json
[cc@karthi-khj059-n01 data]$ vi books.json
```

Below books.json file is displayed after executing the above commands shown in the screen print.

```
["milton-paradise.txt", "[ Paradise Lost by John Milton 1667 ] Book I Of Man ' s
 first disobedience , and the fruit Of that forbidden tree whose mortal taste Br
ought death into the World , and all our woe , With loss of Eden , till one grea
ter Man Restore us , and regain the blissful seat , Sing , Heavenly Muse , that
, on the secret top Of Oreb , or of Sinai , didst inspire That shepherd who firs
t taught the chosen seed In the beginning how the heavens and earth Rose out of
Chaos : or , if Sion hill Delight thee more , and Siloa ' s brook that flowed Fa
st by the oracle of God , I thence Invoke thy aid to my adventurous song , That
with no middle flight intends to soar Above th ' Aonian mount , while it pursues
 Things unattempted yet in prose or rhyme ."]
["edgeworth-parents.txt", "[ The Parent ' s Assistant , by Maria Edgeworth ] THE
 ORPHANS . Near the ruins of the castle of Rossmore , in Ireland , is a small ca
bin , in which there once lived a widow and her four children . As long as she w
as able to work , she was very industrious , and was accounted the best spinner
in the parish ; but she overworked herself at last , and fell ill , so that she
could not sit to her wheel as she used to do , and was obliged to give it up to
her eldest daughter , Mary ."]
```

Open invert_index.py file to view the map and reducer program which is used to generate the output file.

**Defaultdict:** whenever you need a dictionary, and each element's value should start with a default value, so use a **defaultdict**.

**Mapper:** Document identifier content is stored in key and the document content is stored to value.

It uses for loop for the words and if loop to match the words from the different txt files.

**Reducer:** It lists the word and the txt file which contains the key word.

```
import MapReduce
import sys
from collections import defaultdict

mr = MapReduce.MapReduce()

def mapper(record):
    # key: document identifier
    # value: document contents
    key = record[0]
    value = record[1]
    words = value.split()
    words_seen = defaultdict(int)
    for w in words:
      if words_seen[w] != 1:
        mr.emit_intermediate(w, key)
        words_seen[w] += 1

def reducer(key, list_of_values):
    # key: word
    # value: list of occurrence counts
    mr.emit((key, list_of_values))

if __name__ == '__main__':
  inputdata = open(sys.argv[1])
  mr.execute(inputdata, mapper, reducer)
```

**Output of the Invert_index file:**

To generate the output file we need to execute the below command

```
[cc@karthi-khj059-n01 data]$ cd ..
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ ls
assymetric_friendships.py  friends_count.py  MapReduce.py   README.md
data                       invert_index.py   MapReduce.pyc  solutions
dna.py                     join.py           multiply.py    wordcount.py
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ vi invert_index.py
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ python invert_index.py data/book
s.json
```

**Output file:** The output file displays the word and its corresponding text file.

```
["small", ["edgeworth-parents.txt"]]
["ETYMOLOGY", ["melville-moby_dick.txt"]]
["disposition", ["austen-emma.txt"]]
["mistress", ["austen-emma.txt"]]
["beginning", ["milton-paradise.txt", "bible-kjv.txt"]]
["En", ["whitman-leaves.txt"]]
["created", ["bible-kjv.txt"]]
["Being", ["shakespeare-caesar.txt"]]
["could", ["edgeworth-parents.txt"]]
["keep", ["whitman-leaves.txt"]]
["Cob", ["shakespeare-caesar.txt"]]
["place", ["austen-emma.txt"]]
["mortal", ["milton-paradise.txt"]]
["consequence", ["austen-emma.txt"]]
["castle", ["edgeworth-parents.txt"]]
["fallen", ["austen-emma.txt"]]
["first", ["milton-paradise.txt", "bible-kjv.txt", "whitman-leaves.txt"]]
["Lamb", ["blake-poems.txt"]]
["there", ["edgeworth-parents.txt", "bible-kjv.txt"]]
```

# Friends Count:

In this program each key is a person and each value is a friend of the person. This program mainly counts number of friends each person has.

```
[cc@karthi-khj059-n01 data]$ ls
books.json  dna.json  friends.json  matrix.json  records.json
[cc@karthi-khj059-n01 data]$ vi friends.json
```

The input is a two element list: [personA, personB] where

**personA** - Name of a person formatted as a string

**personB** - Name of the personA's friend formatted as a string.

The output of the program should be (person, friend count) where person is a string and the friend count is an integer describing number of friends that the person has.

**Input file( friends.json):** Input file contains the list of friends names.

```
["Myriel", "Geborand"]
["Myriel", "Champtercier"]
["Myriel", "Count"]
["Myriel", "OldMan"]
["Myriel", "Valjean"]
["Napoleon", "Myriel"]
["MlleBaptistine", "Myriel"]
["MlleBaptistine", "Valjean"]
["MlleBaptistine", "MmeMagloire"]
["MmeMagloire", "Myriel"]
["Champtercier", "Myriel"]
["Valjean", "Myriel"]
["Valjean", "MmeMagloire"]
["Valjean", "Labarre"]
["Valjean", "Marguerite"]
["Valjean", "MmeDeR"]
["Valjean", "Isabeau"]
["Valjean", "Fantine"]
["Valjean", "Cosette"]
["Valjean", "Simplice"]
["Valjean", "Woman1"]
["Valjean", "Judge"]
["Valjean", "Woman2"]
"friends.json" 27L, 669C
```

Open friends_count.py file to view the code used to get the count of the friends.

```
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ ls
assymetric_friendships.py  friends_count.py  MapReduce.py   README.md
data                       invert_index.py   MapReduce.pyc  solutions
dna.py                     join.py           multiply.py    wordcount.py
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ vi friends_count.py
```

**Mapper:** Key stores value of the personA's name who has friend and value stores the friends count(PersonA).

**Reducer:** mr.emit displays the personA name and the count of friends personA has.

```
import MapReduce
import sys
from collections import defaultdict

mr = MapReduce.MapReduce()

def mapper(record):
    # key: personA who has a friend
    # value: friend count
    person_with_friend = record[0]
    mr.emit_intermediate(person_with_friend, 1)

def reducer(key, list_of_values):
    # key: person
    # value: friend count

    #count how many friends
    friend_count = len(list_of_values)
    mr.emit((key, friend_count))

if __name__ == '__main__':
  inputdata = open(sys.argv[1])
  mr.execute(inputdata, mapper, reducer)
```

**Output:** Execute the below command to display the results

```
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ python friends_count.py data/fri
ends.json
```

Output file displays the person name and the count of friends.

```
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ python friends_count.py data/fri
"MlleBaptistine", 3]
"Myriel", 5]
"Valjean", 16]
"MmeMagloire", 1]
"Champtercier", 1]
"Napoleon", 1]
```

# Asymmetric Friendship:

In this program we implemented MapReduce algorithm to generate a list of all non symmetric friend relationship.

The input is a two element list: [personA, personB] where

**personA** - Name of a person formatted as a string

**personB** - Name of the personA's friend formatted as a string.

```
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ ls
assymetric_friendships.py  data  dna.py  friends_count.py  invert_index.py  join
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ vi assymetric_friendships.py
```

**Mapper:** record[0] value is stored to person_a and record[1] value is stored to person_b where the final output of the mapper displays both (friend_a, friend_b) list along with(friend_b, friend_a) list.

cc@karthi-khj059-n01:~/hadoop/mapreduce/MapReduce-Basics-master

```
import MapReduce
import sys
from collections import defaultdict

mr = MapReduce.MapReduce()

# =============================
# Do not modify above this line

def mapper(record):
    # key: personA who has a friend
    # value: friend count
    person_a = record[0]
    person_b = record[1]
    mr.emit_intermediate(person_a + person_b, record)
    r_record = [record[1], record[0]]
    mr.emit_intermediate(person_b + person_a, r_record)

def reducer(key, list_of_values):
    # key: person
    # value: friend count

    #count how many friends
    relationship_count = len(list_of_values)
    if relationship_count == 1:
      mr.emit((list_of_values[0][0],list_of_values[0][1]))

# Do not modify below this line
# =============================
if __name__ == '__main__':
  inputdata = open(sys.argv[1])
  mr.execute(inputdata, mapper, reducer)
```

Open friends.json file to view the input folder, where the json file displays the list of friends.

```
[cc@karthi-khj059-n01 data]$ ls
books.json  dna.json  friends.json  matrix.json  records.json
[cc@karthi-khj059-n01 data]$ vi friends.json
```

By executing above command it displays the list of friends data which is used to execute the specified algorithm.

```
["Myriel", "Geborand"]
["Myriel", "Champtercier"]
["Myriel", "Count"]
["Myriel", "OldMan"]
["Myriel", "Valjean"]
["Napoleon", "Myriel"]
["MlleBaptistine", "Myriel"]
["MlleBaptistine", "Valjean"]
["MlleBaptistine", "MmeMagloire"]
["MmeMagloire", "Myriel"]
["Champtercier", "Myriel"]
["Valjean", "Myriel"]
["Valjean", "MmeMagloire"]
["Valjean", "Labarre"]
["Valjean", "Marguerite"]
["Valjean", "MmeDeR"]
["Valjean", "Isabeau"]
["Valjean", "Fantine"]
["Valjean", "Cosette"]
["Valjean", "Simplice"]
["Valjean", "Woman1"]
["Valjean", "Judge"]
["Valjean", "Woman2"]
"friends.json" 27L, 669C
```

Execute the program by giving the below command by providing file which contain the program and the input data file which is used for processing.

```
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ ls
assymetric_friendships.py  data  dna.py  friends_count.py  invert_index.py  join
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ vi assymetric_friendships.py
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ vi assymetric_friendships.py
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ python assymetric_friendships.py
 data/friends.json
```

After executing the above program it displays the asymmetric relation values both (friend_a, friend_b) and (friend_b , friend_a).

```
[cc@karthi-khj059-n01 data]$ cd ..
[cc@karthi-khj059-n01 MapReduce-Basics-master]$ python assymetric_friendships.py
["Myriel", "Count"]
["Valjean", "Babet"]
["Myriel", "Napoleon"]
["Labarre", "Valjean"]
["Valjean", "Gillenormand"]
["MlleBaptistine", "MmeMagloire"]
["MlleBaptistine", "Valjean"]
["MlleGillenormand", "Valjean"]
["Babet", "Valjean"]
["MmeDeR", "Valjean"]
["Count", "Myriel"]
["Valjean", "Simplice"]
["Geborand", "Myriel"]
["Judge", "Valjean"]
["Marguerite", "Valjean"]
["Valjean", "Woman1"]
["Valjean", "Woman2"]
["Montparnasse", "Valjean"]
["Myriel", "Geborand"]
["MmeMagloire", "Valjean"]
["Myriel", "MlleBaptistine"]
["Valjean", "Cosette"]
```