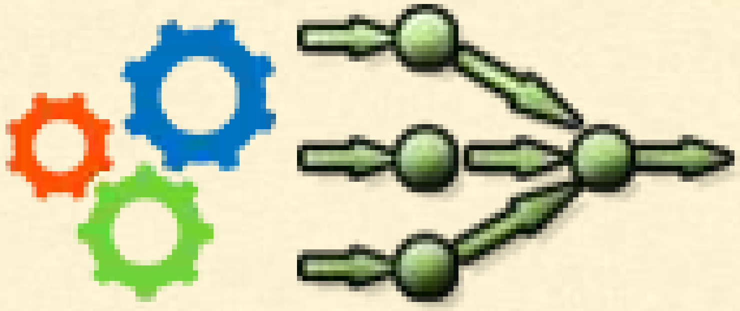


Welcome to Session on Hadoop Streaming

Why Hadoop Streaming?

It is a Hadoop Library which makes it possible to use *any binary* as mapper or reducer

Why?

- Java mapreduce is cumbersome
- Legacy code as mapper or reducer
- Many non-java programmers

Why is not Hadoop Streaming?

- Real time data processing
- Continously running a process

  (Unbounded Data Processing)

MAP / REDUCE

Streaming Job

A Hadoop Library which makes it possible to use *any binary* as mapper or reducer

Mapper - gives out key<tab>value per line

MAP / REDUCE

Streaming Job

A Hadoop Library which makes it possible to use *any binary* as mapper or reducer

Mapper - gives out key<tab>value per line

Reducer - gets ungrouped sorted data key<tab>value

MAP / REDUCE

Streaming Job

A Hadoop Library which makes it possible to use *any binary* as mapper or reducer

Mapper - gives out key<tab>value per line
Reducer - gets ungrouped sorted data key<tab>value

```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar
-input /data/mr/wordcount/input -output wordcount_output_unix
-mapper 'sed "s/ /\n/g"' -reducer "uniq -c"
```

This is our mapper

This is the reducer

MAP / REDUCE

Streaming Job

Word Count using unix commands as mapper & Reducer

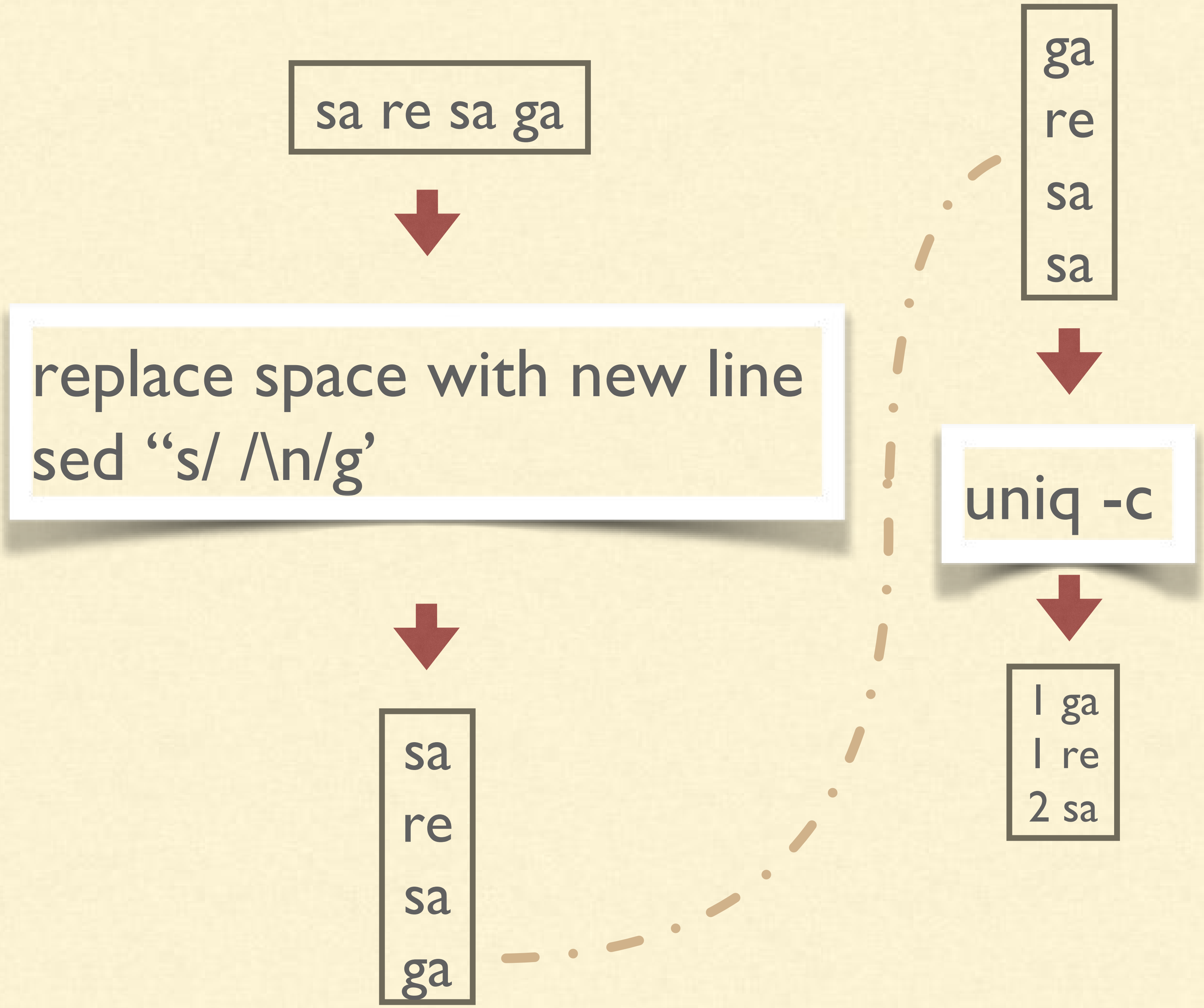
```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar
-input /data/mr/wordcount/input
-output wordcount_output_unix
-mapper 'sed "s/ /\n/g"'
-reducer "uniq -c"
```

This is our mapper

This is the reducer

MAP / REDUCE

Streaming Job



sa re sa ga

replace space with new line
sed “s/ /\n/g’

sa
re
sa
ga

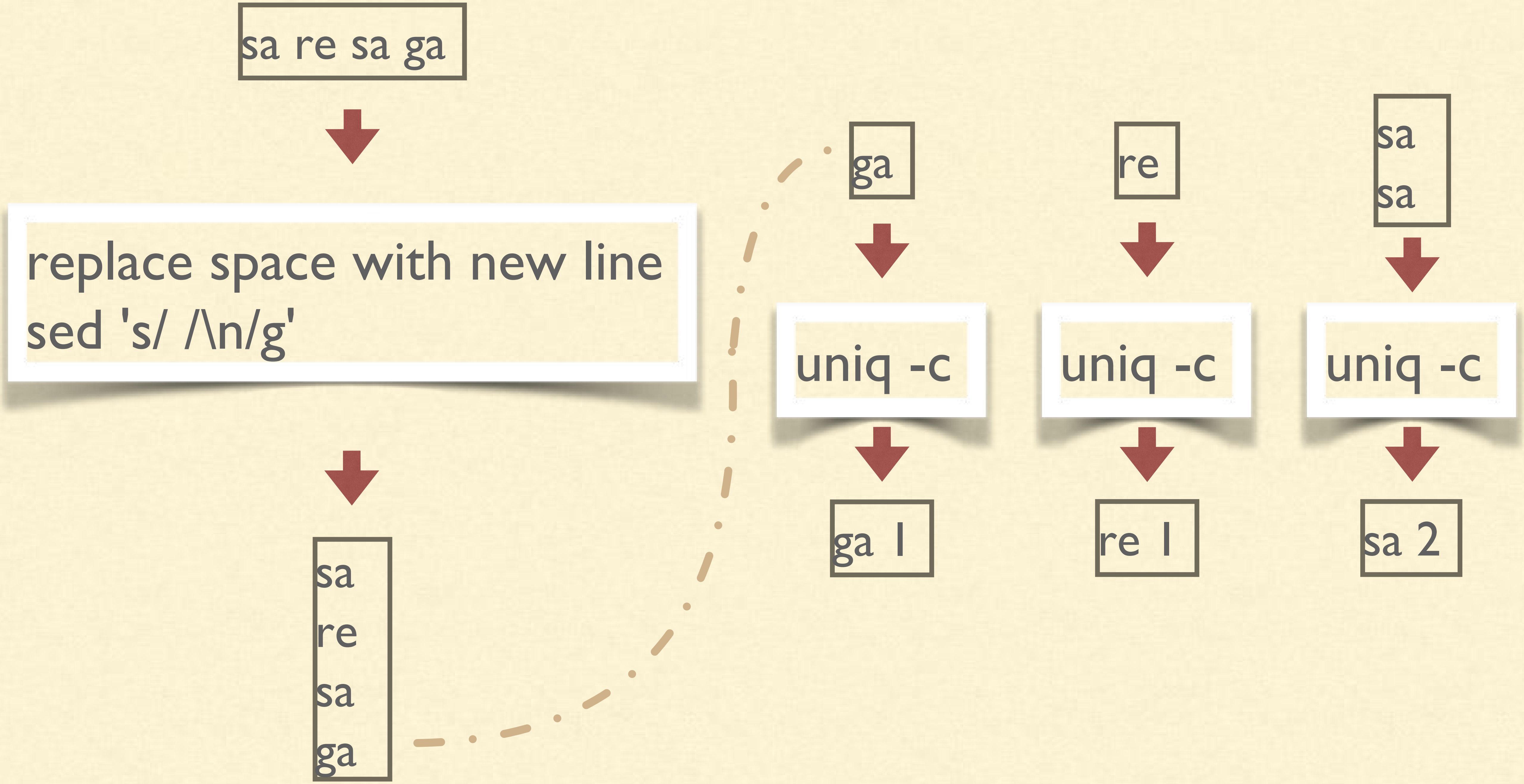
ga
re
sa
sa

uniq -c

1 ga
1 re
2 sa

MAP / REDUCE

Streaming Job - More Reducers



sa re sa ga

replace space with new line
sed 's/ /\n/g'

sa
re
sa
ga

ga

uniq -c

ga 1

re

uniq -c

re 1

sa
sa

uniq -c

sa 2

MAP / REDUCE

Streaming Job

Ship a script

#mycmd.sh - clean up further

```
#!/bin/bash
sed -r 's/[ \t]+/\n/g' | sed "s/[^a-zA-Z0-9]//g" | tr "A-Z" "a-z"
```

```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar
  -D mapred.reduce.tasks=2
  -input /data/mr/wordcount/input/
  -output wordcount_clean_unix
  -mapper ./mycmd.sh
  -reducer 'uniq -c'
  -file mycmd.sh
```

Multiple Reducer Argument: *-D mapred.reduce.tasks=2*

STREAMING JOB - HANDS-ON

Doc:

http://hadoop.apache.org/docs/r1.2.1/streaming.html

MAP / REDUCE

Notes

- OK to have no reducer
  - hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -input sgiri/wordcount/input/ -output sgiri/wordcount/output21fe32/ -mapper ./mycmd.sh -file mycmd.sh
- If no reducer and don’t want sorting
  - use -D mapred.reduce.tasks=0
  - Maps will decide the number of output files
- Number of Maps
  - A function of number of InputSplits
  - conf.setNumMapTasks(int num) or -D mapred.map.tasks=1

MAP / REDUCE

Number of Reduces

More reducer mean

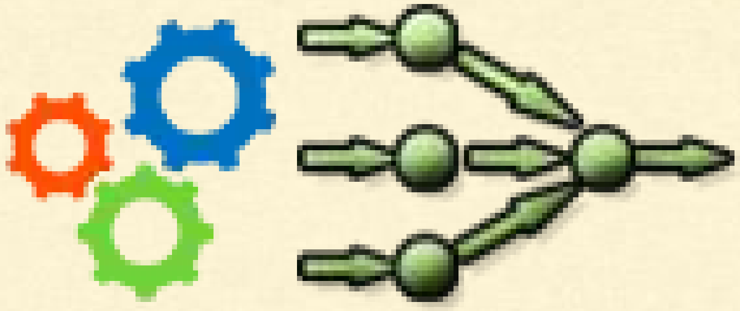
  - Faster
  - More framework load
  - Lowers chances of failure

- (0.95 to 1.75) * (Max Tasks)
- Max Tasks =
  - No. of Nodes * Max Reduce tasks simultaneously per task tracker.
  - mapreduce.tasktracker.reduce.tasks.maximum = 2
-

MAP / REDUCE

Testing

- First test on very small data
  - Random Sample data
- Separately Test Mapper and Reducer
- Steaming Job could be tested with simple unix command:
  - cat inputfile | *mymapper* | sort | my*reducer > outputfile*



Hadoop Streaming

Thank you!

Hadoop & Spark

Thank you.

+1 419 665 3276 (US)
+91 803 959 1464 (IN)

support@knowbigdata.com

Subscribe to our Youtube channel for latest videos -
https://www.youtube.com/channel/UCxugRFe5wETYA7nMH6VGyEA