A decorative L-shaped frame made of thick dark blue lines. One L-shape is in the top-left corner, and another is in the bottom-right corner, framing the central text.

MAPREDUCE FUNDAMENTAL CONCEPTS

Why MapReduce?

- Distributes the processing of data on your cluster
- Divides your data up into partitions that are MAPPED (transformed) and REDUCED (aggregated) by mapper and reducer functions you define
- Resilient to failure - an application master monitors your mappers and reducers on each partition



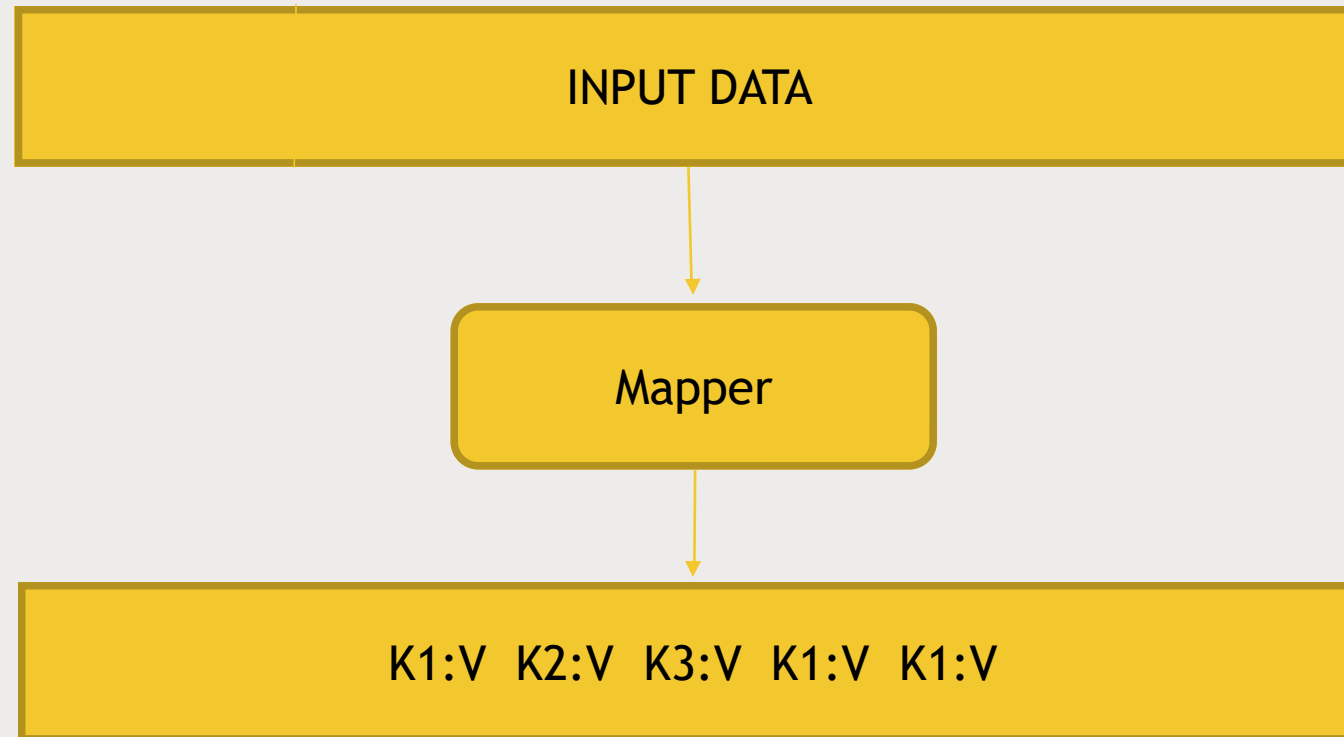
Let's illustrate with an example

- How many movies did each user rate in the MovieLens data set?



How MapReduce Works: Mapping

- The MAPPER converts raw source data into **key/value** pairs



Example: MovieLens Data (u.data file)

USER ID | MOVIE ID | RATING | TIMESTAMP

196 242 3 881250949

186 302 3 891717742

196 377 1 878887116

244 51 2 880606923

166 346 1 886397596

186 474 4 884182806

186 265 2 881171488

Map users to movies they watched

USER ID	MOVIE ID	RATING	TIMESTAMP
---------	----------	--------	-----------

196	242	3	881250949
-----	-----	---	-----------

186	302	3	891717742
-----	-----	---	-----------

196	377	1	878887116
-----	-----	---	-----------

244	51	2	880606923
-----	----	---	-----------

166	346	1	886397596
-----	-----	---	-----------

186	474	4	884182806
-----	-----	---	-----------

186	265	2	881171488
-----	-----	---	-----------

Mapper

196:242 186:302 196:377 244:51 166:346 186:274 186:265

Extract and Organize What We Care About

196:242 186:302 196:377 244:51 166:346 186:274 186:265



MapReduce Sorts and Groups the Mapped Data (“Shuffle and Sort”)

196:242 186:302 196:377 244:51 166:346 186:274 186:265



166:346 186:302,274,265 196:242,377 244:51

The REDUCER Processes Each Key's Values

166:346 186:302,274,265 196:242,377 244:51



len(movies)



166:1 186:3 196:2 244:1

Putting it All Together

USER ID	MID	R	T
---------	-----	---	---

196	242	3	881250949
-----	-----	---	-----------

186	302	3	891717742
-----	-----	---	-----------

196	377	1	878887116
-----	-----	---	-----------

244	51	2	880606923
-----	----	---	-----------

166	346	1	886397596
-----	-----	---	-----------

186	474	4	884182806
-----	-----	---	-----------

186	265	2	881171488
-----	-----	---	-----------



MAPPER



196:242 186:302 196:377 244:51 166:346 186:274 186:265



SHUFFLE AND SORT



166:346 186:302,274,265 196:242,377 244:51



REDUCER



166:1 186:3 196:2 244:1



MAPREDUCE ON A CLUSTER

How MapReduce Scales



Putting it All Together

USER ID	MID	R	T
---------	-----	---	---

196	242	3	881250949
-----	-----	---	-----------

186	302	3	891717742
-----	-----	---	-----------

196	377	1	878887116
-----	-----	---	-----------

244	51	2	880606923
-----	----	---	-----------

166	346	1	886397596
-----	-----	---	-----------

186	474	4	884182806
-----	-----	---	-----------

186	265	2	881171488
-----	-----	---	-----------

MAPPER

196:242 186:302

196:377 244:51

166:346 186:274 186:265

SHUFFLE AND SORT

166:346 186:302,274,265

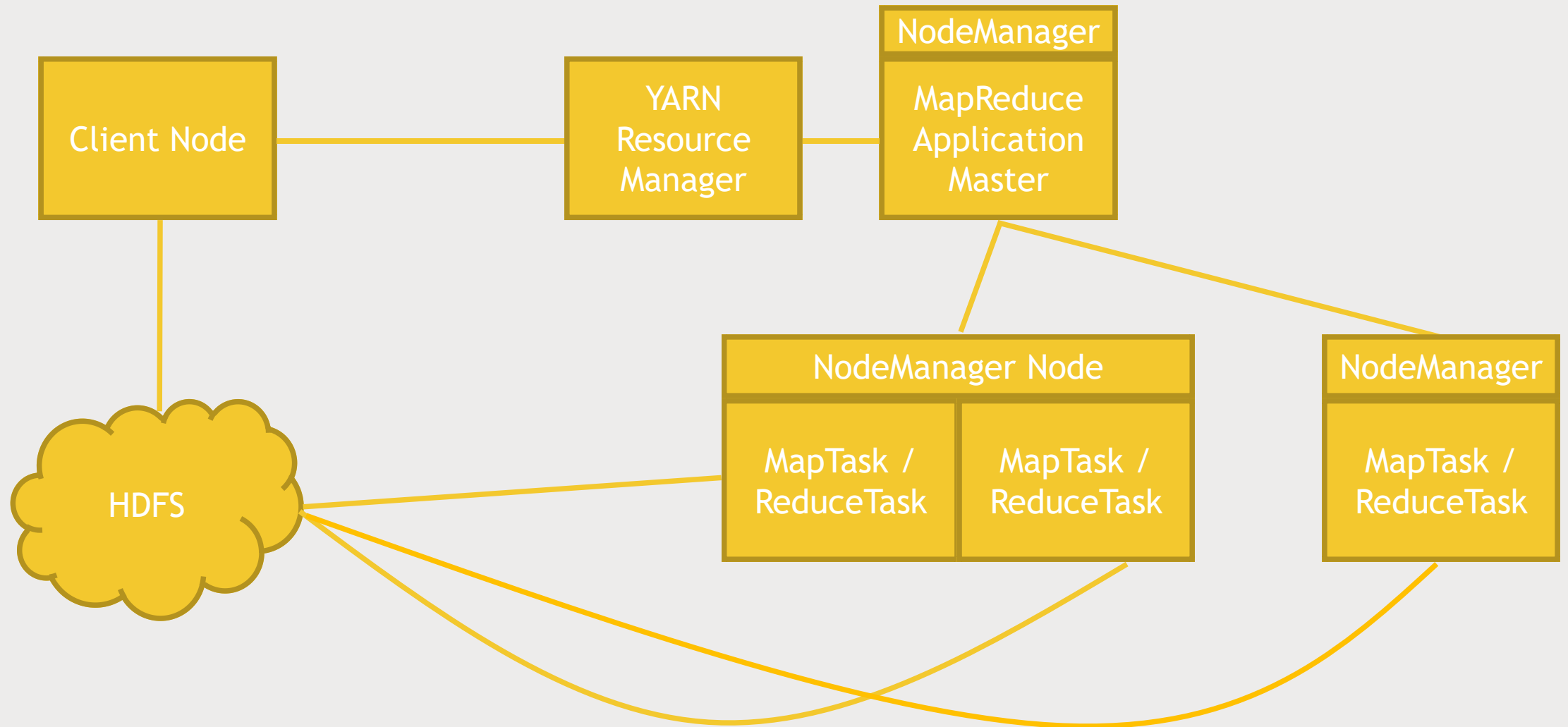
196:242,377 244:51

REDUCER

166:1 186:3

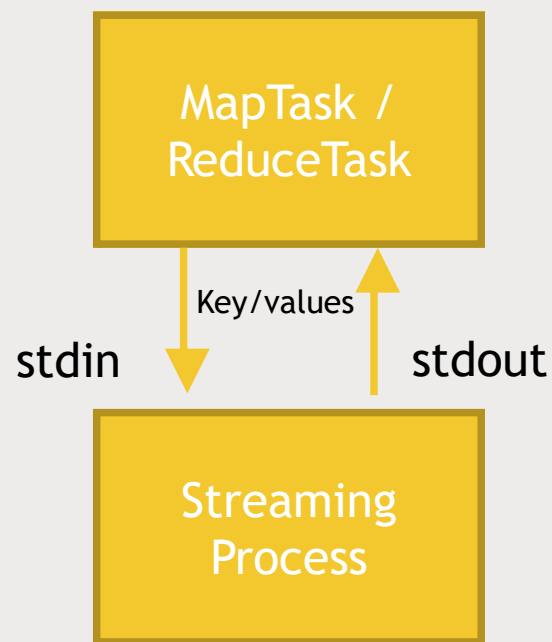
196:2 244:1

What's Happening



How are mappers and reducers written?

- MapReduce is natively Java
- STREAMING allows interfacing to other languages (ie Python)




Handling Failure



- Application master monitors worker tasks for errors or hanging
 - *Restarts as needed*
 - *Preferably on a different node*
- What if the application master goes down?
 - *YARN can try to restart it*
- What if an entire Node goes down?
 - *This could be the application master*
 - *The resource manager will try to restart it*
- What if the resource manager goes down?
 - *Can set up “high availability” (HA) using Zookeeper to have a hot standby*





MAPREDUCE: A REAL EXAMPLE

How many of each rating type exist?



How many of each movie rating exist?



Making it a MapReduce problem

- MAP each input line to (rating, 1)
- REDUCE each rating with the sum of all the 1's

USER ID|MOVIE ID|RATING|TIMESTAMP

196 242 3 881250949

186 302 3 891717742

196 377 1 878887116

244 51 2 880606923

166 346 1 886397596

186 474 4 884182806

186 265 2 881171488

Map

3,1

3,1

1,1

2,1

1,1

4,1

2,1

Shuffle
& Sort

1 -> 1, 1

2 -> 1, 1

3 -> 1, 1

4 -> 1

Reduce

1, 2

2, 2

3, 2

4, 1

Writing the Mapper

USER ID|MOVIE ID|RATING|TIMESTAMP

196 242 3 881250949
186 302 3 891717742
196 377 1 878887116
244 51 2 880606923
166 346 1 886397596
186 474 4 884182806
186 265 2 881171488

Map

3,1
3,1
1,1
2,1
1,1
4,1
2,1

Shuffle
& Sort

1 -> 1, 1
2 -> 1, 1
3 -> 1, 1
4 -> 1

Reduce

1, 2
2, 2
3, 2
4, 1

```
def mapper_get_ratings(self, _, line):  
    (userID, movieID, rating, timestamp) = line.split('\t')  
    yield rating, 1
```

Writing the Reducer

USER ID|MOVIE ID|RATING|TIMESTAMP

196 242 3 881250949
186 302 3 891717742
196 377 1 878887116
244 51 2 880606923
166 346 1 886397596
186 474 4 884182806
186 265 2 881171488

Map

3,1
3,1
1,1
2,1
1,1
4,1
2,1

Shuffle
& Sort

1 -> 1, 1
2 -> 1, 1
3 -> 1, 1
4 -> 1

Reduce

1, 2
2, 2
3, 2
4, 1

```
def reducer_count_ratings(self, key, values):  
    yield key, sum(values)
```

Putting it all together


```
from mrjob.job import MRJob
from mrjob.step import MRStep

class RatingsBreakdown(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ratings,
                  reducer=self.reducer_count_ratings)
        ]

    def mapper_get_ratings(self, _, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield rating, 1

    def reducer_count_ratings(self, key, values):
        yield key, sum(values)

if __name__ == '__main__':
    RatingsBreakdown.run()
```



RUNNING MAPREDUCE WITH MRJOB

Run our MapReduce job in our Hadoop installation



Installing what we need

- PIP

- *Utility for installing Python packages*
- `su root`
`yum install python-pip`

- Nano

- `yum install nano`

- MRJob

- `pip install mrjob`
`exit`

- Data files and the script

- `wget http://media.sundog-soft.com/hadoop/ml-100k/u.data`
- `wget http://media.sundog-soft.com/hadoop/RatingsBreakdown.py`

Running with mrjob

- Run locally
 - *python RatingsBreakdown.py u.item*
- Run with Hadoop
 - *python MostPopularMovie.py -r hadoop --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar u.data*



YOUR CHALLENGE

Sort movies by popularity with Hadoop



Challenge exercise

- Count up ratings given for each movie
 - *All you need is to change one thing in the mapper - we don't care about ratings now, we care about movie ID's!*
 - *Start with this and make sure you can do it.*
 - *You can use nano to just edit the existing RatingsBreakdown.py script*

Stretch goal

- Sort the movies by their numbers of ratings
- Strategy:
 - *Map to (movieID, 1) key/value pairs*
 - *Reduce with output of (rating count, movieID)*
 - *Send this to a second reducer so we end up with things sorted by rating count!*
- Gotchas:
 - *How do we set up more than one MapReduce step?*
 - *How do we ensure the rating counts are sorted properly?*

Multi-stage jobs

- You can chain map/reduce stages together like this:

```
def steps(self):  
    return [  
        MRStep(mapper=self.mapper_get_ratings,  
            reducer=self.reducer_count_ratings),  
        MRStep(reducer=self.reducer_sorted_output)  
    ]
```

Ensuring proper sorting

- By default, streaming treats all input and output as strings. So things get sorted as strings, not numerically.
- There are different formats you can specify. But for now let's just zero-pad our numbers so they'll sort properly.

- The second reducer will look like this:

```
def reducer_count_ratings(self, key, values):  
    yield str(sum(values)).zfill(5), key
```

Iterating through the results

- Spoiler alert!

```
def reducer_sorted_output(self, count, movies):  
    for movie in movies:  
        yield movie, count
```

CHECK YOUR RESULTS

Did it work?



My solution

```
from mrjob.job import MRJob
from mrjob.step import MRStep

class RatingsBreakdown(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ratings,
                   reducer=self.reducer_count_ratings),
            MRStep(reducer=self.reducer_sorted_output)
        ]

    def mapper_get_ratings(self, _, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield movieID, 1

    def reducer_count_ratings(self, key, values):
        yield str(sum(values)).zfill(5), key

    def reducer_sorted_output(self, count, movies):
        for movie in movies:
            yield movie, count

if __name__ == '__main__':
    RatingsBreakdown.run()
```




HDFS



YARN



MapReduce2



Tez



Hive



HBase



Pig



Sqoop



Oozie



ZooKeeper



Falcon



Storm



Flume



Ambari Infra



Ambari Metrics



Atlas



Kafka



Knox



Ranger



Spark



Spark2

Summary

Heatmaps

Configs

Quick Links ▾

Summary

[NameNode](#)

Started

No alerts

Disk Remaining 23.2 GB / 42.0

[SNameNode](#)

Stopped

No alerts

Blocks (total) 797

[DataNodes](#)

1/1 Started

Block Errors 0 corrupt replica / 0 missing / 15 under replicated

DataNodes Status 1 live / 0 dead / 0 decommissioning

Total Files + Directories 1110

[NFSGateways](#)

0/1 Started

Upgrade Status No pending upgrade

NameNode Uptime 168.13 secs

Safe Mode Status Not in safe mode

NameNode Heap 68.0 MB / 240.0 MB (28.3% used)

Disk Usage (DFS Used) 1.7 GB / 42.0 GB (4.05%)

Disk Usage (Non DFS Used) 17.1 GB / 42.0 GB (40.77%)

YARN Queue Manager

Files View

Hive View

Pig View

Storm View

Tez View

Metrics

Actions ▾

Last 1 hour ▾

NameNode GC count

No Data No available data
for the time period.

NameNode GC time

No Data No available data
for the time period.

NN Connection Load

No Data No available data
for the time period.

NameNode Heap

No Data No available data
for the time period.

NameNode Host Load

No Data No available data
for the time period.

NameNode RPC

Failed disk volumes

Blocks With Corrupted
Replicas

Under Replicated Blocks

HDFS Space Utilization



Total: 12 files or folders

+ Select All

New Folder

Upload



Search in current directory...



Name >	Size >	Last Modified >	Owner >	Group >	Permission
mr-history	--	2016-10-25 03:48	mapred	hadoop	drwxrwxrwx
app-logs	--	2016-10-25 04:10	yarn	hadoop	drwxrwxrwx
ats	--	2016-10-25 03:48	yarn	hadoop	drwxr-xr-x
demo	--	2016-10-25 04:01	hdfs	hdfs	drwxr-xr-x
hdp	--	2016-10-25 03:48	hdfs	hdfs	drwxr-xr-x
mapred	--	2016-10-25 03:48	mapred	hdfs	drwxr-xr-x
apps	--	2016-10-25 03:54	hdfs	hdfs	drwxr-xr-x
ranger	--	2016-10-25 03:47	hdfs	hdfs	drwxr-xr-x
spark-history	--	2016-11-11 15:50	spark	hadoop	drwxrwxrwx
spark2-history	--	2016-10-25 04:14	spark	hadoop	drwxrwxrwx
tmp	--	2016-10-25 04:11	hdfs	hdfs	drwxrwxrwx
user	--	2016-10-25 04:11	hdfs	hdfs	drwxr-xr-x



/ > user > maria_dev

Total: 3 files or folders

+ Select All

New Folder

Upload



Add new folder



Name

ml-100k

Cancel

+ Add

Name >



.Trash

files-view

hive

--

2016-11-09 12:21

maria_dev

Group >

Permission

hdfs

drwx-----

hdfs

drwxr-xr-x

hdfs

drwxr-xr-x



/ > user > maria_dev > ml-100k

Total: 0 files or folders

+ Select All

New Folder

Upload



1

Name >



Upload file to /user/maria_dev/ml-100k



Drag file to upload or click to browse

Currently supports single file upload

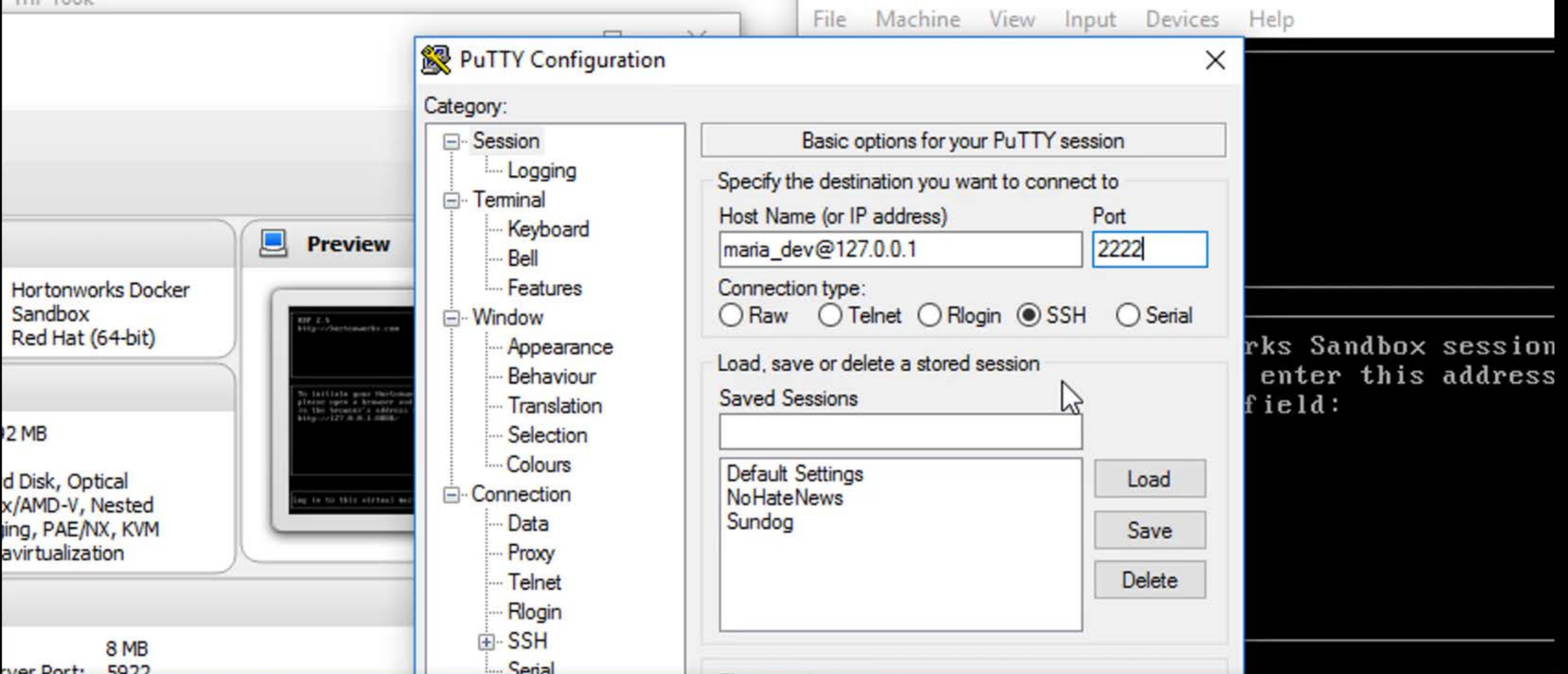
Cancel

Search in current directory...



Group >

Permission



On MacOS or Linux, open a terminal and run:
`ssh maria_dev@127.0.0.1 -p 2222`

Using username "maria_dev".

maria_dev@127.0.0.1's password:

Last login: Fri Nov 11 20:39:50 2016 from 10.0.2.2

[maria_dev@sandbox ~]\$ hadoop fs -ls

Found 3 items

drwx-----	- maria_dev	hdfs	0	2016-11-11	20:53	.Trash
drwxr-xr-x	- maria_dev	hdfs	0	2016-11-11	20:16	files-view
drwxr-xr-x	- maria_dev	hdfs	0	2016-11-09	17:21	hive

[maria_dev@sandbox ~]\$ hadoop fs -mkdir ml-100k

[maria_dev@sandbox ~]\$ hadoop fs -ls

Found 4 items

drwx-----	- maria_dev	hdfs	0	2016-11-11	20:53	.Trash
drwxr-xr-x	- maria_dev	hdfs	0	2016-11-11	20:16	files-view
drwxr-xr-x	- maria_dev	hdfs	0	2016-11-09	17:21	hive
drwxr-xr-x	- maria_dev	hdfs	0	2016-11-11	20:58	ml-100k

[maria_dev@sandbox ~]\$

```
[maria_dev@sandbox ~]$ wget http://media.sundog-soft.com/hadoop/ml-100k/u.data
--2016-11-11 20:59:41--  http://media.sundog-soft.com/hadoop/ml-100k/u.data
Resolving media.sundog-soft.com... 54.231.72.51
Connecting to media.sundog-soft.com|54.231.72.51|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2079229 (2.0M) [application/octet-stream]
Saving to: "u.data"

100%[=====>] 2,079,229    3.64M/s    in 0.5s

2016-11-11 20:59:41 (3.64 MB/s) - "u.data" saved [2079229/2079229]

[maria_dev@sandbox ~]$
```

```
[maria_dev@sandbox ~]$ ls
```

```
u.data
```

```
[maria_dev@sandbox ~]$ ls -la
```

```
total 2056
```

```
drwx----- 1 maria_dev maria_dev 4096 Nov 11 20:59 .
drwxr-xr-x 1 root      root      4096 Oct 25 08:04 ..
-rw----- 1 maria_dev maria_dev  896 Nov 11 20:39 .bash_history
-rw-r--r-- 1 maria_dev maria_dev   18 May 10  2016 .bash_logout
-rw-r--r-- 1 maria_dev maria_dev  176 May 10  2016 .bash_profile
-rw-r--r-- 1 maria_dev maria_dev  124 May 10  2016 .bashrc
-rw-rw-r-- 1 maria_dev maria_dev 2079229 Nov 11 20:32 u.data
```

```
[maria_dev@sandbox ~]$ hadoop fs -copyFromLocal u.data ml-100k/u.data
```

```
[maria_dev@sandbox ~]$
```



```
[maria_dev@sandbox ~]$ hadoop fs -copyFromLocal u.data ml-100k/u.data
[maria_dev@sandbox ~]$ hadoop fs -ls ml-100k
Found 1 items
-rw-r--r--    1 maria_dev hdfs      2079229 2016-11-11 21:00 ml-100k/u.data
[maria_dev@sandbox ~]$ hadoop fs -rm ml-100k/u.data
16/11/11 21:00:59 INFO fs.TrashPolicyDefault: Moved: 'hdfs://sandbox.hortonworks
.com:8020/user/maria_dev/ml-100k/u.data' to trash at: hdfs://sandbox.hortonworks
.com:8020/user/maria_dev/.Trash/Current/user/maria_dev/ml-100k/u.data
[maria_dev@sandbox ~]$
```

```
[maria_dev@sandbox ~]$ hadoop fs -rmkdir ml-100k
```

```
[maria_dev@sandbox ~]$ hadoop fs -ls
```

```
Found 3 items
```

```
drwx----- - maria_dev hdfs          0 2016-11-11 20:53 .Trash
```

```
drwxr-xr-x - maria_dev hdfs          0 2016-11-11 20:16 files-view
```

```
drwxr-xr-x - maria_dev hdfs          0 2016-11-09 17:21 hive
```

```
[maria_dev@sandbox ~]$
```



```
[maria_dev@sandbox ~]$ hadoop fs
Usage: hadoop fs [generic options]
    [-appendToFile <localsrc> ... <dst>]
    [-cat [-ignoreCrc] <src> ...]
    [-checksum <src> ...]
    [-chgrp [-R] GROUP PATH...]
    [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
    [-chown [-R] [OWNER][:[GROUP]] PATH...]
    [-copyFromLocal [-f] [-p] [-l] <localsrc> ... <dst>]
    [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
    [-count [-q] [-h] [-v] [-t [<storage type>]] <path> ...]
    [-cp [-f] [-p | -p[topax]] <src> ... <dst>]
    [-createSnapshot <snapshotDir> [<snapshotName>]]
    [-deleteSnapshot <snapshotDir> <snapshotName>]
    [-df [-h] [<path> ...]]
    [-du [-s] [-h] <path> ...]
    [-expunge]
    [-find <path> ... <expression> ...]
    [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
    [-getfacl [-R] <path>]
    [-getfattr [-R] {-n name | -d} [-e en] <path>]
    [-getmerge [-nl] <src> <localdst>]
    [-help [cmd ...]]
    [-ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [<path> ...]]
```

Using username "maria_dev".

maria_dev@127.0.0.1's password:

[maria_dev@sandbox ~]\$ su root

Password:

You are required to change your password immediately (root enforced)

Changing password for root.

(current) UNIX password:

New password:

Retype new password:

[root@sandbox maria_dev]# yum install python-pip

Loaded plugins: fastestmirror, ovl, priorities

Setting up Install Process

Determining fastest mirrors

epel/metalink		17 kB	00:00
---------------	--	-------	-------


```
from mrjob.job import MRJob
from mrjob.step import MRStep

class RatingsBreakdown(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ratings,
                  reducer=self.reducer_count_ratings)
        ]

    def mapper_get_ratings(self, _, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield rating, 1

    def reducer_count_ratings(self, key, values):
        yield key, sum(values)

if __name__ == '__main__':
    RatingsBreakdown.run()
```

[Read 19 lines (Converted from DOS format)]