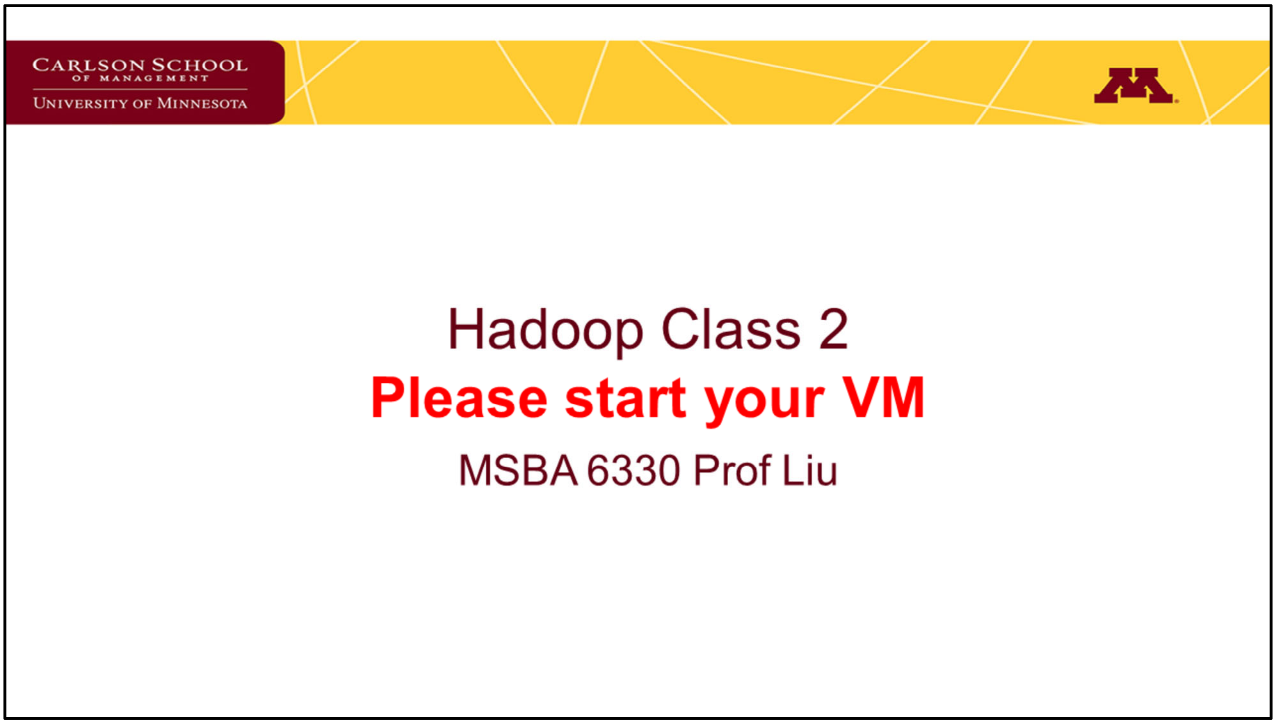
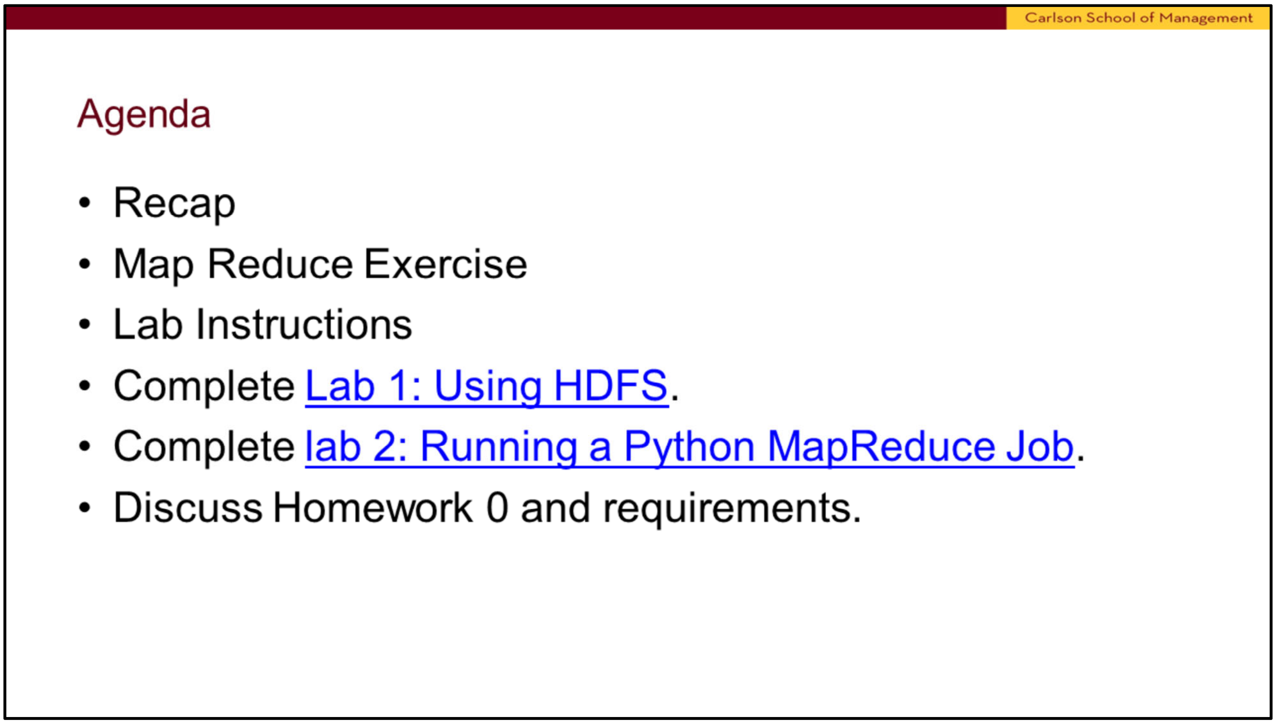


Hadoop Class 2

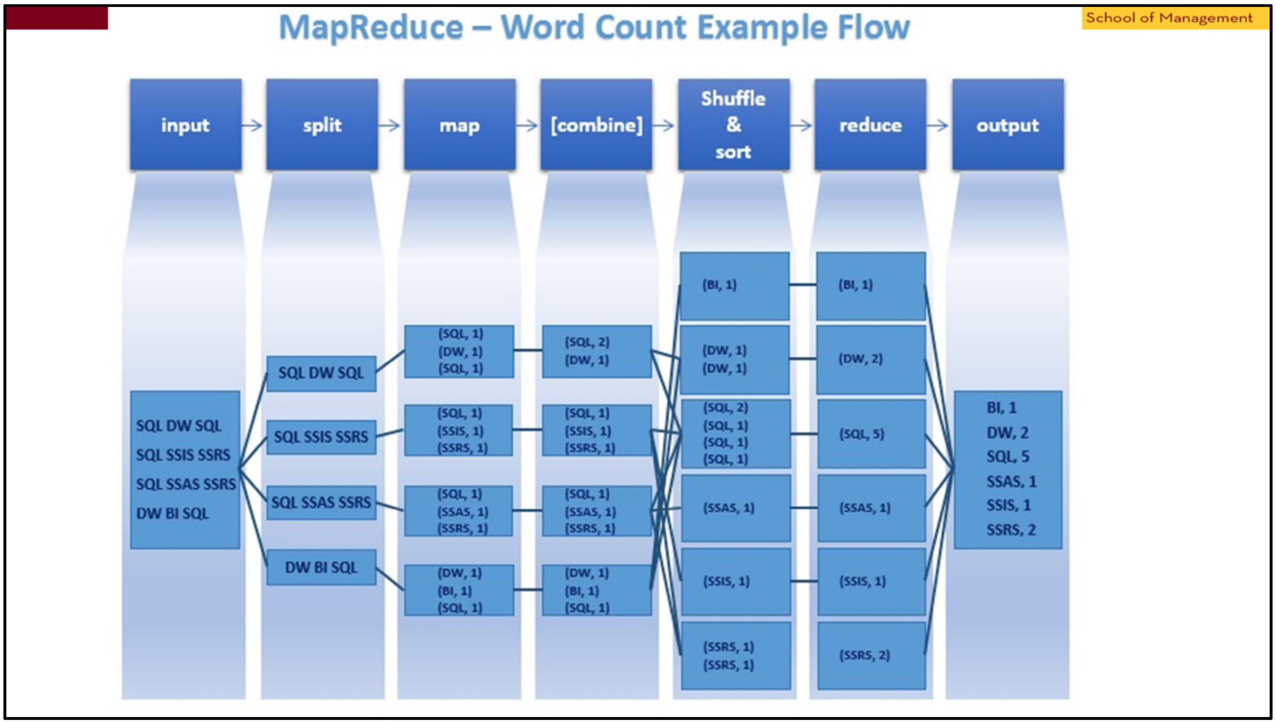
Please start your VM

MSBA 6330 Prof Liu



Agenda

- Recap
- Map Reduce Exercise
- Lab Instructions
- Complete Lab 1: Using HDFS.
- Complete lab 2: Running a Python MapReduce Job.
- Discuss Homework 0 and requirements.



MapReduce – Word Count Example Flow

School of Management

input → split → map → [combine] → Shuffle & sort → reduce → output

SQL DW SQL
SQL SSIS SSRS
SQL SSAS SSRS
DW BI SQL

SQL DW SQL
SQL SSIS SSRS
SQL SSAS SSRS
DW BI SQL

(SQL, 1)
(DW, 1)
(SQL, 1)

(SQL, 1)
(SSIS, 1)
(SSRS, 1)

(SQL, 1)
(SSAS, 1)
(SSRS, 1)

(DW, 1)
(BI, 1)
(SQL, 1)

(SQL, 2)
(DW, 1)

(SQL, 1)
(SSIS, 1)
(SSRS, 1)

(SQL, 1)
(SSAS, 1)
(SSRS, 1)

(DW, 1)
(BI, 1)
(SQL, 1)

(BI, 1)

(DW, 1)
(DW, 1)

(SQL, 2)
(SQL, 1)
(SQL, 1)
(SQL, 1)

(SSAS, 1)

(SSIS, 1)

(SSRS, 1)
(SSRS, 1)

(BI, 1)

(DW, 2)

(SQL, 5)

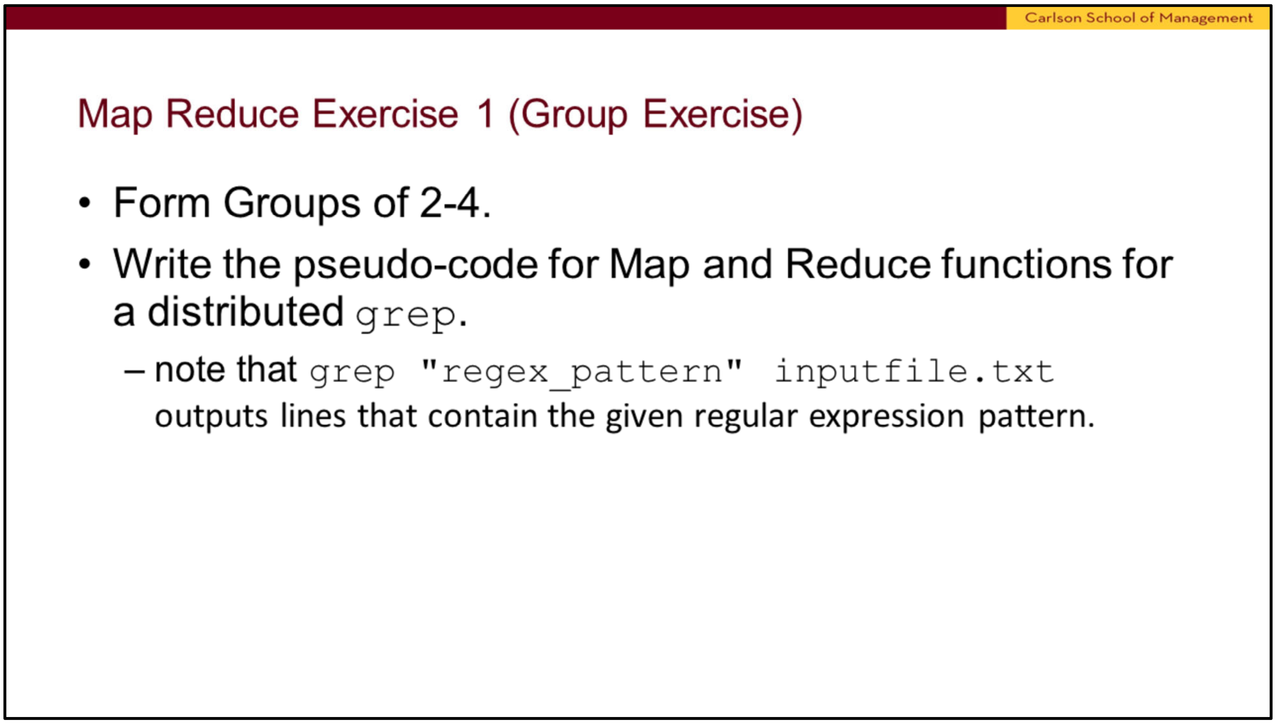
(SSAS, 1)

(SSIS, 1)

(SSRS, 2)

BI, 1
DW, 2
SQL, 5
SSAS, 1
SSIS, 1
SSRS, 2

```
map(String key, String value):
key: line number
value: line content
for each word w in value:
  EmitIntermediate(word, "1");

reduce(String key, Iterator values):
key: a word
values: a list of counts
int result = 0;
for each v in values:
     result += Int(v);
Emit (key,result)
```



Map Reduce Exercise 1 (Group Exercise)

- Form Groups of 2-4.
- Write the pseudo-code for Map and Reduce functions for a distributed `grep`.
  - note that `grep "regex_pattern" inputfile.txt` outputs lines that contain the given regular expression pattern.

the map function emits a line if it matches a supplied regular expression pattern.
The reduce function is an identify function that just copies the supplied intermediate data to the output.

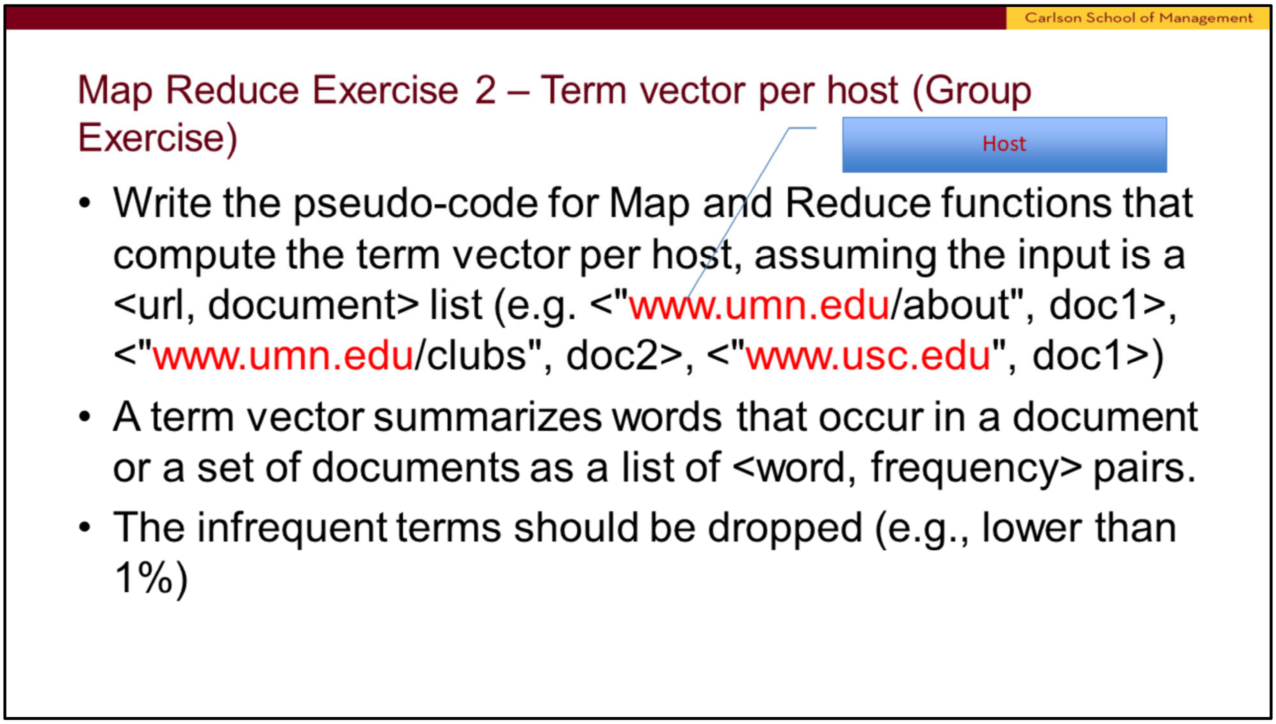
```
map(key, value)
# key: is the line number
# value: is a line in the document e.g. "The quick brown fox jumps over the lazy dog...."

if the line matches the given pattern then:
    output: line
```

We don't need a reducer for this; this is a map only job.

Alternatively, you can use an identity reducer

```
map(key, value)
# key: is a line
# values: a list of Nones (one for each instance of the line)
```

```
for each value in values:
    output: value
```



Carlson School of Management

Map Reduce Exercise 2 – Term vector per host (Group Exercise)

Host

- Write the pseudo-code for Map and Reduce functions that compute the term vector per host, assuming the input is a <url, document> list (e.g. <"www.umn.edu/about", doc1>, <"www.umn.edu/clubs", doc2>, <"www.usc.edu", doc1>)
- A term vector summarizes words that occur in a document or a set of documents as a list of <word, frequency> pairs.
- The infrequent terms should be dropped (e.g., lower than 1%)

Map function emits a <hostname, term vector> for each input document, where hostname is extract from the URL of the document.
the Reduce function is passed all per-document term vectors for a given host.
It adds these term vectors, throwing away infrequent terms, and then emits a final <hostname, term vector> pair.

You can assume in the input comes in (url,document) pairs
www.umn.edu/about: "The quick brown fox jumps over the lazy dog...."
www.umn.edu/support: "Donor support helps U of M researchers learn more about which brain aneurysms need treatment....."
www.umn.edu/contact-us: "Contact Us Twin Cities Campus Information: 612-625-5000"
www.usc.edu/about: "...."
www.usc.edu/transportation: "...."

```
map(key, value)
# key: is the url for the document
# value: is the actual document.
```

```
extract host from url  #e.g. www.umn.edu
Obtain term vector from the document #e.g., [("fox",1),("quick",1),("lazy",10),...]
output (host, term vector)  #e.g., (www.umn.edu, [("fox",1),("quick",1),("lazy",1),...])

reduce(key,values)
# key: is a host
# values: is a list (iterator) of term vectors, e.g. [("fox",1),("quick",1),("the",10),...],
[("support",1),("Donor",1),("the",3),...]

combine the term vectors into one, e.g.
[("fox",1),("quick",1),("support",1),("Donor",1),("the",13),...]
output (host, combined term vector)
```