## Fall 2016

Department of Computer Science University of Helsinki

Big Data Management

**General Instruction**: The computer science department has a high performance cluster, **Ukko**, which is available to all department's users. You can check the list of nodes and their status from <u>www.cs.helsinki.fi/ukko/hpc-report.txt</u>. Public nodes are available using CS user accounts and work exactly like CS department's interactive Linux servers. In this course, you can run your Hadoop Mapreduce codes on Ukko to answer the exercise questions.

**Step 1** : If your machine runs Windows, then use WinSCP (https://winscp.net/eng/index.php) to connect to the Ukko server. If you are using linux or Mac OS, please use the built-in SSH command

Sa Login - WinSCP		<b>— — X</b>		
Wew Site	Session Elle protocol: SFTP Host name: ukko026.hpc.cs.helsnio.fi User name:	Port number:		
Iools V Manage V	Save V	Advanced 🖣		

If you can't login in the Ukko nodes, please either

(1) login to melkinpaasi.cs.helsinki.fi via SSH, then connect to Ukko using SSH from this server, or

(2) Use HY-VPN OpenVPN (this is important, Pulse Secure is NOT able to connect Ukko): https://helpdesk.it.helsinki.fi/en/instructions/logging-and-connections/networks/connections-outside-university

**Step 2** : Access to the folder */cs/work/scratch*, build a new folder for yourself, move the Hadoop binary file here (drag it to the right-part window). In this document, we use *Hadoop.2.6.5.tar.gz* 

🖕 chzhang - chzhang@u	ukko026.hpc.cs.helsinki.fi - Win	SCP					l	- • • ×
Local Mark Files Co	mmands Session Options F	Remote Help						
册 🚟 🚔 Synchronize	e 🗖 🥐 💽 🛞 🕅 🕯	Queue • Transfer Settings De	fault ·	<i>6</i> •				
chzhang@ukko026.h	hpc.cs.helsinki.fi 🚅 New Ses	sion						
K. C: OSDisk	• 🖪 🔽 🗠 • + + + 1	🖬 🖬 🏠 🎜 🗞		🔒 chzhang 🔹 🔗 🔽 🔍	⊨ • -> - i 🖬 🖡	🖬 🏫 🍠 👒 Find File	s Ben	
	ia 🖋 🧷 De Deservation 🎮			E da Sa la	/ D. Descetion		-	
Bill obioso Bill N Eo					an up roperties			
L:\				/cs/work/scratch/chzhang				
Name	Size Type	Changed		Name	Size	Changed	Rights	Owner
🕌 HY-Data	File folder	12.9.2016 10:46:52		📥 e.		27.10.2016 13:44:17	rwxrwxrwt	root
🍌 Intel	File folder	28.7.2015 14:16:48		line hadoop-2.6.5		28.10.2016 15:05:29	rwx	chzhang
PerfLogs	File folder	14.7.2009 6:20:08		hadoop-2.6.5.tar.gz	194 957 KB	27.10.2016 13:47:28	rw	chzhang
🍌 Perl	File folder	28.7.2015 14:48:31						
Program Files	File folder	3.10.2016 19:14:52						
🎍 Program Files (x86)	File folder	26.10.2016 9:58:40						
Python33	File folder	28.7.2015 14:49:29						
鷆 ssm	File folder	28.7.2015 14:33:06						
퉬 swsetup	File folder	15.1.2016 15:31:51						
🍌 tmp	File folder	26.10.2016 10:41:19						
🍌 Users	File folder	27.9.2016 16:26:23						
퉬 Windows	File folder	19.10.2016 9:40:45						
D (0.D) 0 (12			101-11					
B OT U B IN U OF 12			10 hidden	0 B OL TAT MIR IN 0 OL 5		0		0.05.25
						II	SELK-3	2 0:06:21

Step 3 : Open the Putty embedded in the WinSCP, and unzip the Hadoop file.



The command is tar xvf Hadoop.2.6.5.tar.gz

If Putty is not installed in your machine, please download and install it.

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

**Step 4** : Edit the Hadoop configure file **Hadoop-env.sh** under Hadoop-2.6.5/etc/hadoop (Right-click the file in the WinSCP and edit directly)

Add following the lines:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

**Step 5** : Now the Hadoop environment is ready, next we show how to compile a WordCount MapReduce program and create a jar in the server. Download the source code of MapReduce here:

<u>https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-</u> <u>core/MapReduceTutorial.html</u>

Move the MapReduce source codes into the folder Hadoop-2.6.5/.

Then use putty to input the following commands

Step 6 : Compile the java file

\$ bin/hadoop com.sun.tools.javac.Main WordCount.java

Create the jar file

\$ jar cf wc.jar WordCount\*.class

#### Create the input folder

\$ mkdir input

### Create two files f1 and f2 under input folder

Hello World Bye World

and

Hello Hadoop Goodbye Hadoop

#### Run WordCount code

\$ bin/hadoop jar wc.jar WordCount wordcount/input wordcount/output

Assume that the output folder does not exist here.

## View the result

\$ bin/hadoop fs -cat wordcount/output/part-r-00000

# If success, you should see the following output:

Bye 1

Goodbye 1

Hadoop 2

Hello 2

World 2