

# CS 496/522 Cloud Computing Programming Assignment 1

Huiying Chen  
800722249

Note: Detailed walk through written in Prog1\_readme.md. This document is a supplement for task tracking, results showcase, and for easy grading.

## 1. Setting up VMs – 20 pts

- Deploy 4 nodes (1 light and 3 heavy) with specs; (done)
- Deploy Ubuntu-server (v.22) on all four nodes (done)
- Do the OS and system setup
  - Setup the IP address configuration; create and establish connectivity between the nodes (I have renamed my cluster nodes to be hadoop-master, hadoop-worker1, hadoop-worker2, and hadoop-worker3)

```
(base) hyingchen@hyings-MacBook-Pro: ~ % ssh admin@128.123.168.205
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Apr 14 09:28:27 PM UTC 2023

System load:  0.0      Processes:      215
Usage of /:   84.6% of 115.84GB   Users logged in: 1
Memory usage: 24%      IPv4 address for ens160: 192.168.202.205
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.
https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

2 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Apr 14 15:18:16 2023 from 128.123.30.77
admin@hadoop-master: ~$

admin@hadoop-master:~$ ssh admin@hadoop-worker1
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Apr 14 09:29:03 PM UTC 2023

System load:  0.0      Processes:      234
Usage of /:   29.4% of 57.77GB   Users logged in: 1
Memory usage: 14%      IPv4 address for ens160: 192.168.202.206
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.
https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

41 updates can be applied immediately.
3 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Apr 13 20:00:18 2023 from 192.168.202.205
admin@hadoop-worker1:~$
```

```

admin@hadoop-master:~$ ssh admin@hadoop-worker2
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Apr 14 09:32:27 PM UTC 2023

System load:  0.0      Processes:      233
Usage of /:   29.4% of 57.7GB    Users logged in: 1
Memory usage: 11%      IPv4 address for ens160: 192.168.202.207
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

41 updates can be applied immediately.
3 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Apr 14 21:32:27 2023 from 192.168.202.205
admin@hadoop-worker2:~$

admin@hadoop-master:~$ ssh admin@hadoop-worker3
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Apr 14 09:33:38 PM UTC 2023

System load:  0.0068359375    Processes:      233
Usage of /:   19.8% of 57.7GB    Users logged in: 1
Memory usage: 10%      IPv4 address for ens160: 192.168.202.208
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

27 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Apr 14 21:33:39 2023 from 192.168.202.205
admin@hadoop-worker3:~$

```

- Setup firewall protection; Open Firewall access from 128.123.63.0/24, 128.123.64.0/24, and 10.253.0.0/16 over ssh to your master node.

```

admin@hadoop-master:~$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
Anywhere ALLOW 128.123.63.0/24
Anywhere ALLOW 128.123.64.0/24
22 ALLOW 10.253.0.0/16
22/tcp (v6) ALLOW Anywhere (v6)

```

- Create user named admin and give them sudo access. Add two ssh keys (one for Gaurav and another for Sharad) as authorized keys for the admin user:

```

admin@hadoop-master:~$ id admin
uid=1001(admin) gid=1001(admin) groups=1001(admin)
admin@hadoop-master:~$ sudo usermod -aG sudo admin
[sudo] password for admin:
admin@hadoop-master:~$ sudo adduser admin sudo
The user 'admin' is already a member of 'sudo'.

```

```
-- admin@hadoop-master: ~ -- ssh admin@128.123.160.205

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC1s0pKp4QdVhZLXQU88g046kURJ9iBEhhJu785u6Emx0IGJxrSZnh74qswZcnHUM2dEd8iAfjhAL/P2e05EMB0qn4x1g2r82
ZIGpIMfE8lG42TXAJYb8xG7XzfVdceHThEuEs+HAAxbLYKHxWnta+riAirtXf4JiSdXu+D/N1Yen4VyK9cuUZWQIEj97/0ZkxXtVixt0eE/ToyEKj04bp1REzBREMSsq6ZKhr
P1oWHGUZ/0ntfs6C+8BYBCfH2RRLzzGYAK+J3UpLtl+UWLRp1eM//7z32LQZ8Xyia7oSLxCXfY24ITN1JnubYH05WG0GKL0M/Akf0wFdd5v gpanwar@nsol1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC1s7PhiG8e7L4g6FPNpEub8VD2379IVglvFSmkpx5PFPR91FT7W2f0PGXxcwvu0aZAKGwjCIZlqGnMG9QHLsZ6Lxt0JFSd+4/
gm2Df1gzuoHplA6e30S06j2B51Zj36vPMdpdDudc2FYBYfcQQeH2CxxoDDesw7SVnuLNaMMWfWwxaZz5g5vmoAYoUT0qavtFNJAoxYKpkvaw7m+ZTiMk2hKBfAy03PCXvR9k3
X122R+gZH1GKzZ9cFj0K2h/KIIoKwwXY7KCI00ySTGLptNpyvIs0+5UydcplcnjFvJJNU84f5VQVg0L39k2IYXzbbF6GqynoTzG6x1f4bJcCUDsK0tNAuZX971C8aow/x+RubGh6I
ALXL26nUwrTnjwogBnbsl/S6ytdk7ZpniT82Wd5qdmKqXdcZB0PHIujdkEgNrsbDLuXxEl72dMzqJpQgfbWuw8DI0TRa04pKDLugvCiLAYjw1FBL7TzDvWZVXBrcICfjshpZvuh/
BUTV2U=theshrestha@theshrestha-desktop
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC+kpdReD9ISgIW213J+1UpzIVfXy7ssaTw3zI2NFEgpmat60VYH90FSAF9iMzbTFR86Ye+opba123SF0M+8YamxVP6FNAmIQHl
6Re9mttJumXC0sAzSRegkgmdRZFyZ1fAnrn4k+AgR6RDaKtbQa0pWx0MhktarDPZ1n2X179yIVVqpYQdacggNk9UYKcsGsd3eB+fbLcgHDarmk1i0TgV1RWzmAKnDzplCoH/y
xBOQ7RPP/3dAV18c61c02C1cGAXkFH8co6FAM77M1r6XBePVCBSBM0B6L7de3yC3f/1qNMhzyt4BVM7mu4rjuTY+wKM7poKstTEqVpPybZMcEsMcw0I6BM4gryKCwoxBehEEQ0
F7JlBkBRelyLZ9u0qSak14RAfewkvGf1FCVgQGVvN7jyvmQs6Gn0o0Yox41DYM2T86MdQqYVRsXkJHAcudTyFSwU4QQWfJk4T5x8ZHufctmuGuGe2Kh5vzKr68gLV//8G4Yo
6NF8Q10= hyingchen@hyings-MacBook-Pro.local
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQCDSKoicded0z4VQ/Y2CSDBKehC4AA1T070+HPVIMPCZnWnE1tILDNxxkVYM4woirndN40NFIGDGLUPwGvFYJ791beWIKZ11eVR
XJABdxYBoZx9s0CCn+5GYaq070PYe9GEmOfT1S5yRW54r+6udx8Uc5/BMHMaTyEbK6Rd9hjiMwYJVAQmomHrUuiMRoyV0tMN5xhkRmhAFJ+Ptu7ZyqXZvGVMEaw0WeTBiAdHj
0IHTVQ0QHvMHDZnZ/J9mN4R8TmVHa3PCc1XCcvRLmw27dZHisMNUzqN16+kPc+NRdbpr4te+ze4e47mkRix9dNJKZUcFP93w2VAz6ad8U905yQKZn6iNM6iSVR08z8A58G0n4nby
xo///GiWY4tupiAzHnExzq48DRG0c0K9JNg/Vg1tptmupFPBNLJzKpJnLaGg53LR1WQYAwEKRRK/cymZbgusCyhVgsUwVbjbeJmCvYtkGv/1tN7yja9Mn5C05BkPncuyW5CT1uZ
9tM5izM= admin@supervisor
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQCgHTTa5bJdYVcIcqX9M1d5AVXUQyF/fjTbKaJY9VgKQWDX9Zg7G3Tiy0niQpcCEKZ9IqBpH+jHBE6YE8/hUPD9kpQwFvz0EFp
Xv1lggf88sT5ALrU7RsJ/7RPCMIbXNn3461rR0iTeZxGtrU/0ApPhth/8u7t4L1A/2z2Tdxj0cc58ubqRrKwXRXeUM4J01hK0wH8Xwz0zS1Vetu1lfrj4TQuik3XfMSId8GET
vs7B1a/0x9udjebJfF/DUcT7EEQcsYAIgaFuMoQrW0+7r8h9h155T1TWUSyzEx0TKSiCWfXbEg1Cja3Ws2437uyZApP/YeZaWcY18RHISaU+PvQtdj2baZx2kPT5P51r0a1g4V
2IxWZkqV6K7bNzbztHgUDzt+bsZnWMDZicnpKbPdPwn/qgxv7vEass+eJd+hz/zBv87ZnZyJ001KIKyuJ380EICrmdFXR86BznXoC7yz6rLYLMdwOwrqNdoqtCyC1PEZ0I+dGX
D2V0eZ0= admin@hadoop-master
```

## 2. Setting up Apache Hadoop and Spark frameworks – 20 pts

```
admin@hadoop-master:~$ jps
14435 SecondaryNameNode
14198 NameNode
14631 ResourceManager
59676 Jps
9340 Master

admin@hadoop-worker1:~$ jps
9747 NodeManager
23082 Jps
9595 DataNode
7916 Worker

admin@hadoop-worker2:~$ jps
7106 Worker
21864 Jps
8574 DataNode

admin@hadoop-worker3:~$ jps
11104 NodeManager
9451 Worker
21085 Jps
```

```
hyingchen — admin@hadoop-master: ~ -- ssh admin@128.123.160.205 — 136x40

-- admin@hadoop-master: ~ -- ssh admin@128.123.160.205

top - 03:20:49 up 9 days, 12:14, 3 users, load average: 1.75, 0.65, 0.56
Tasks: 221 total, 1 running, 220 sleeping, 0 stopped, 0 zombie
%Cpu(s): 40.1 us, 2.8 sy, 0.0 ni, 16.3 id, 40.1 wa, 0.0 hi, 0.7 si, 0.0 st
MiB Mem : 7950.0 total, 122.0 free, 2379.2 used, 5448.8 buff/cache
MiB Swap: 4096.0 total, 4051.3 free, 44.7 used, 5263.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 62987 admin     20   0 3980628 719588 28880 S  83.7   8.8   1:25.76 java
    92 root       20   0      0      0      0 S   1.0   0.0   0:29.68 kswapd0
  9340 admin     20   0 3681212 328420 28636 S   0.3   4.0   3:21.54 java
 14198 admin     20   0 3954532 459156 29192 S   0.3   5.6  13:51.36 java
 14631 admin     20   0 4115344 462416 28772 S   0.3   5.7  20:40.19 java
 55055 root       20   0      0      0      0 I   0.3   0.0   0:06.16 kworker/1:3-mm_percpu_wq
 63073 admin     20   0  10616   4288  3436 R   0.3   0.1   0:00.08 top
    1 root       20   0 167648 12200  7440 S   0.0   0.1   0:06.56 systemd
    2 root       20   0      0      0      0 S   0.0   0.0   0:00.04 kthreadd
    3 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
    6 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
    8 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
```

## 3. Performing the map-reduce and spark tasks – 30 pts; Clear and well-presented output – 10 pts

- Retrieve data available using scp



```

admin@hadoop-master:~$ ls stock_data/
'1. Data description.txt'
ACC_minute_data_with_indicators.csv
ADANIENT_minute_data_with_indicators.csv
ADANIGREEN_minute_data_with_indicators.csv
ADANIPORTS_minute_data_with_indicators.csv
AMBUJACEM_minute_data_with_indicators.csv
APOLLOHOSP_minute_data_with_indicators.csv
ASIANPAINT_minute_data_with_indicators.csv
AUROPHARMA_minute_data_with_indicators.csv
AXISBANK_minute_data_with_indicators.csv
BAJAJ-AUTO_minute_data_with_indicators.csv
BAJAJFINSV_minute_data_with_indicators.csv
BAJAJHLING_minute_data_with_indicators.csv
BAJAJFINSV_minute_data_with_indicators.csv
BANDHANBHK_minute_data_with_indicators.csv
BANKBARODA_minute_data_with_indicators.csv
BERGEPAINT_minute_data_with_indicators.csv
BHARTIARTL_minute_data_with_indicators.csv
BIOCON_minute_data_with_indicators.csv
BOSCHLTD_minute_data_with_indicators.csv
BPCL_minute_data_with_indicators.csv
BRITANNIA_minute_data_with_indicators.csv
CHOLAFIN_minute_data_with_indicators.csv
CIPLA_minute_data_with_indicators.csv
COALINDIA_minute_data_with_indicators.csv
COLPAL_minute_data_with_indicators.csv
DABUR_minute_data_with_indicators.csv
DIVISLAB_minute_data_with_indicators.csv
DLF_minute_data_with_indicators.csv
DMART_minute_data_with_indicators.csv
DRREDDY_minute_data_with_indicators.csv
EICHERMOT_minute_data_with_indicators.csv
GAIL_minute_data_with_indicators.csv
GLAND_minute_data_with_indicators.csv
GODREJCP_minute_data_with_indicators.csv
GRASIM_minute_data_with_indicators.csv
HAVELLS_minute_data_with_indicators.csv
HCLTECH_minute_data_with_indicators.csv
HDFCAMLTD_minute_data_with_indicators.csv
HDFCBANK_minute_data_with_indicators.csv
HDFCLIFE_minute_data_with_indicators.csv
HDFC_minute_data_with_indicators.csv
HEROMOTOCO_minute_data_with_indicators.csv
HINDALCO_minute_data_with_indicators.csv
HINDPETRO_minute_data_with_indicators.csv
HINDUNILVR_minute_data_with_indicators.csv
ICICI1BANK_minute_data_with_indicators.csv
ICICIGI_minute_data_with_indicators.csv
ICICIPRULI_minute_data_with_indicators.csv
IGL_minute_data_with_indicators.csv
INDIGO_minute_data_with_indicators.csv
INDUSTINDBK_minute_data_with_indicators.csv
INDUSTOWER_minute_data_with_indicators.csv
INFY_minute_data_with_indicators.csv
IOC_minute_data_with_indicators.csv
ITC_minute_data_with_indicators.csv
JINDALSTEL_minute_data_with_indicators.csv
JSWSTEEL_minute_data_with_indicators.csv
JUBLFOOD_minute_data_with_indicators.csv
KOTAKBANK_minute_data_with_indicators.csv
LICI_minute_data_with_indicators.csv
LTI_minute_data_with_indicators.csv
LT_minute_data_with_indicators.csv
LUPIN_minute_data_with_indicators.csv
MARICO_minute_data_with_indicators.csv
MARUTI_minute_data_with_indicators.csv
MCDOWELL-N_minute_data_with_indicators.csv
MM_minute_data_with_indicators.csv
MUTHOOTFIN_minute_data_with_indicators.csv
NAUKRI_minute_data_with_indicators.csv
NESTLEIND_minute_data_with_indicators.csv
NIFTY 50_minute_data_with_indicators.csv
NIFTY BANK_minute_data_with_indicators.csv
NMDC_minute_data_with_indicators.csv
NTPC_minute_data_with_indicators.csv
ONGC_minute_data_with_indicators.csv
PEL_minute_data_with_indicators.csv
PGHH_minute_data_with_indicators.csv
PIDILITIND_minute_data_with_indicators.csv
PIIND_minute_data_with_indicators.csv
PNB_minute_data_with_indicators.csv
POWERGRID_minute_data_with_indicators.csv
RELIANCE_minute_data_with_indicators.csv
SAIL_minute_data_with_indicators.csv
SBICARD_minute_data_with_indicators.csv
SBILIFE_minute_data_with_indicators.csv
SBIN_minute_data_with_indicators.csv
SHREECEM_minute_data_with_indicators.csv
SIEMENS_minute_data_with_indicators.csv
SUNPHARMA_minute_data_with_indicators.csv
TATACONSUM_minute_data_with_indicators.csv
TATAMOTORS_minute_data_with_indicators.csv
TATASTEEL_minute_data_with_indicators.csv
TCS_minute_data_with_indicators.csv
TECHM_minute_data_with_indicators.csv
TITAN_minute_data_with_indicators.csv
TORNTPHARM_minute_data_with_indicators.csv
ULTRACEMCO_minute_data_with_indicators.csv
UPL_minute_data_with_indicators.csv
VEDL_minute_data_with_indicators.csv
WIPRO_minute_data_with_indicators.csv
YESBANK_minute_data_with_indicators.csv

```

- Find the maximum number of transactions in a day for all companies in a user supplied given time window.

>> Logic:

- 1) Read arguments of start date and end date
- 2) Construct dataframe structure
- 3) Read csv files as spark dataframes for all companies then combine them into a single one
- 4) Select necessary columns and stock transaction data between start date and end date
- 5) **Sum 'volume' as 'daily\_trade\_count', groupby 'date'** (extract date info from timestamp as date)
- 6) Find the **max daily\_trade\_count value** and output

>> Code developed in spark\_stock\_max\_trans\_all\_given\_win.py

Running Sample:

```

""spark-submit spark_stock_max_trans_all_given_win.py -start <start_date> -end
<end_date>""

```

```

>> spark-submit spark_stock_max_trans_all_given_win.py -start 01/01/2016 -end
12/31/2016 >spark_output/output_stock_max_trans_all_given_win.log
>> check full log file at attached output files

```

- **Maximum stock deviation is defined as (highest price for a day – lowest price for a day)/lowest price for a day. Find the stock that had the highest maximum stock deviation in a day among all stocks and what was the corresponding value for a given time window.**

>> Logic:

- 1) Read arguments of start date and end date
- 2) Construct dataframe structure

- 3) Read csv files as spark dataframes, extracting company\_name from file\_name, then assign the value as 'company' into the corresponding dataframe
- 4) Combine all dataframes into a single one
- 5) Select necessary columns and stock transaction data between start date and end date
- 6) Calculate '**high**'-'**low**' value as '**deviation**', groupby '**company**', '**date**' (extract date info from timestamp as date)
- 7) Find the max '**deviation**' in a day as '**max\_deviation**'
- 8) Find the max '**max\_deviation**' within time window
- 9) Select corresponding company name and date info with the max '**max\_deviation**' value and output

>> Code developed in spark\_stock\_max\_deviation\_all\_given\_win.py

Running Sample:

```
```spark-submit spark_stock_max_deviation_all_given_win.py -start <start_date> -end
<end_date>```
```

```
>> spark-submit spark_stock_max_deviation_all_given_win.py -start 01/01/2016 -end
12/31/2016 >spark_output/output_stock_max_deviation_all_given_win.log
>> check full log file at attached output files
```

- Find the maximum sell price in a day for a given company in the entire data set (No time window required).

>> Logic:

- 1) Read arguments of company name
- 2) Construct dataframe structure
- 3) Read the csv file of given company as a spark dataframe
- 4) Select necessary columns
- 5) Find the **max 'high' value for the given company**
- 6) Find the corresponding date and output

>> Code developed in spark\_stock\_max\_sell\_by\_given\_company.py

Running Sample:

```
```spark-submit spark_stock_max_sell_by_given_company.py -company
<company_name>```
>> spark-submit spark_stock_max_sell_by_given_company.py -company
ADANIANT >spark_output/output_stock_max_sell_given_company.log
>> check full log file at attached output files
```

## 5. Documentation and README for running and grading the code. - 20 pts.

>> Check attached files