



**BDSE19**

# 大數據分析平台Hadoop架設

第一組：戴澤宇、林穎奇、潘敏政、楊為今、蔡汶吟

❖ 此報告由第一組全體同仁協力完成

# 組員分工表

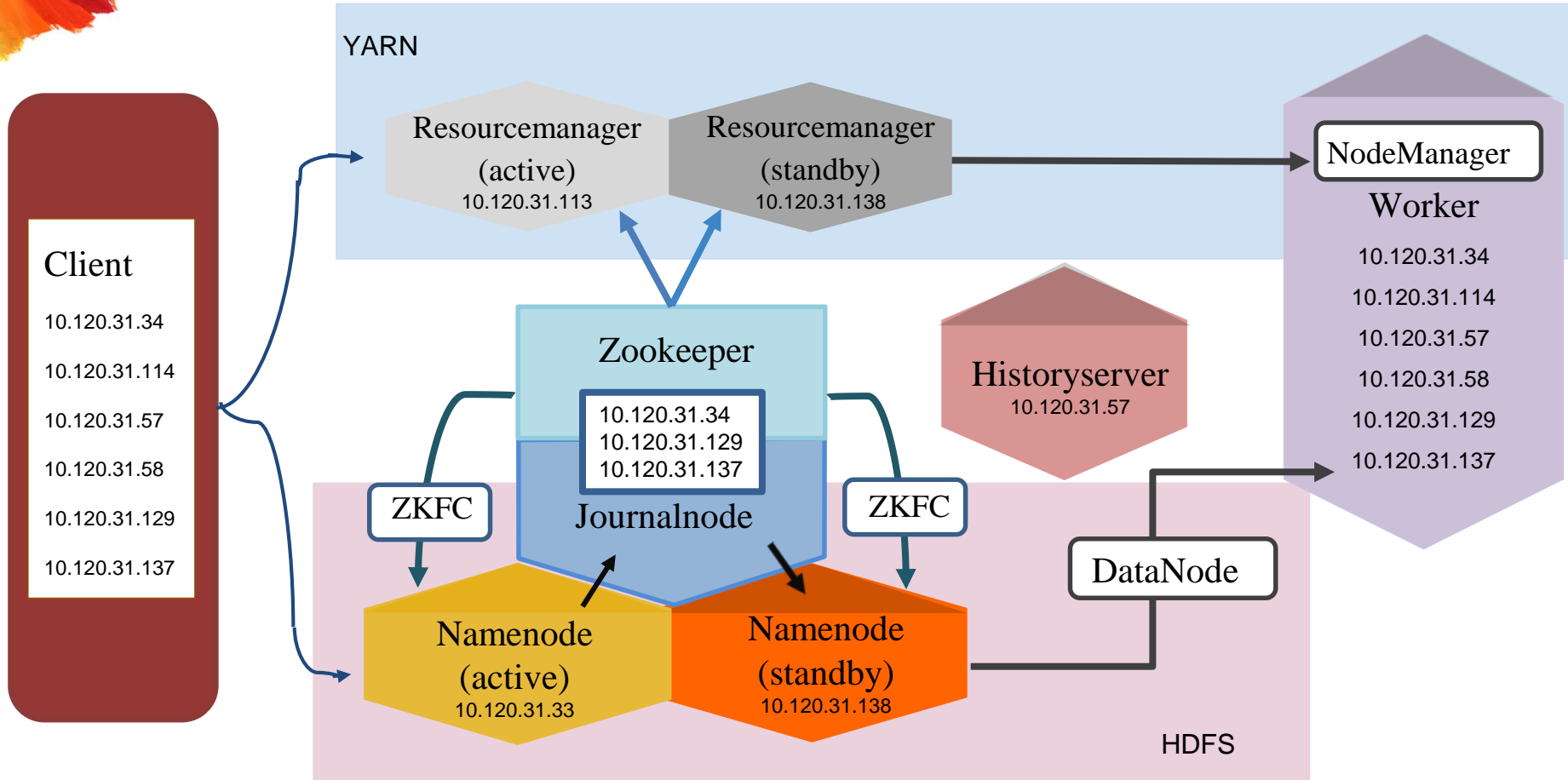
User / Local Host ip	Virtual host	HA clusters
戴澤宇 10.120.31.32	<b>Namenode</b> 10.120.31.33	
	10.120.31.34	<b>Journalnode/Zookeeper</b>
林穎奇 10.120.31.112	<b>Resourcemanager</b> 10.120.31.113	
	10.120.31.114	
楊為今 10.120.31.56	<b>JobHistory</b> 10.120.31.57	
	10.120.31.58	
蔡汶吟 10.120.31.128	10.120.31.129	<b>Journalnode/Zookeeper</b>
	10.120.31.130	<b>Standby Resource Manager</b>
潘敏政 10.120.31.136	10.120.31.137	<b>Journalnode/Zookeeper</b>
	10.120.31.138	<b>Standby Namenode</b>

System Configuration	
Physical host	5
Virtual host	10
CPU	2*10
RAM	8G*10

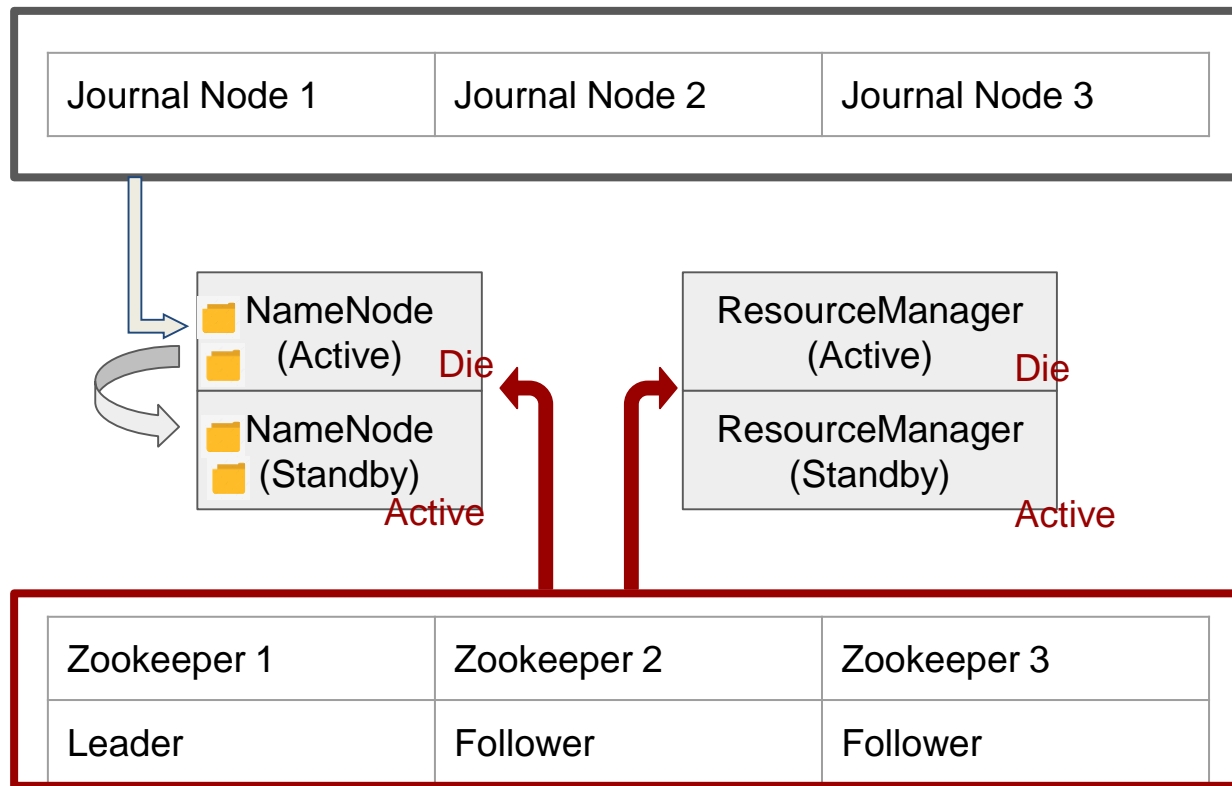
# HADOOP 叢集架構節點說明

NameNode (Master Node)	<p>負責記錄與維護HDFS的Metadata</p> <p>Metadata有：檔案名稱，檔案與其區塊(Blocks)的對應關係，各區塊所在的資料節點(DataNode)，其它的檔案屬性(如：建立時間，複本數量...等)</p>
ResourceManager	管理叢集系統中所有應用程序的資源分配
JobHistoryServer	紀錄執行Map-Reduce各作業的花費時間和啟動、結束時間等詳細資料
DataNode (Slave Node)	<p>檔案區塊(Blocks)實際儲存的地方，通常有多部DataNode</p> <p>DataNode會定期傳送現有的Blocks狀態給NameNode，若NameNode發現某個Block之複本數量少於現有的備份設定時，NameNode會自動增加該Block的複本。</p> <p>當某個DataNode掛掉時，NameNode會自動將此DataNode上的所有Blocks重新配置到其它DataNode上</p>
NodeManager	負責監控各container資源使用情況(cpu、記憶體)，回報給RM
JournalNode	為了同步兩個NameNode，standby NameNode從這獲取NameNode備份的資料
Zookeeper	為分佈式應用提供一致性服務的軟件，提供的功能包括：配置維護、域名服務、分佈式同步、組服務等。

# HADOOP.HA



# High Availability (Zookeeper & JournalNode)



ZKFailoverController  
能及時檢測到 NameNode  
的健康狀況，在主  
NameNode、  
ResourceManager 故障時  
藉助 Zookeeper 實現自動  
的主備選舉和切換

# NN HA

```
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ hdfs haadmin -getServiceState nn1
active
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ hdfs haadmin -getServiceState nn2
standby
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ hdfs haadmin -getServiceState nn1
2021-07-11 12:10:09,523 INFO ipc.Client: Retrying connect to server: bdse33.example.org/10.120.31.33:8020. Already tried 0 time(s); retry policy is RetryUpTo
ximumCountWithFixedSleep(maxRetries=1, sleepTime=1000 MILLISECONDS)
Operation failed: Call From bdse34.example.org/10.120.31.34 to bdse33.example.org/10.120.31.33:8020 failed: java.net.ConnectException: Connection
Refused; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ hdfs haadmin -getServiceState nn2
active
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ hdfs haadmin -getServiceState nn2
active
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ hdfs haadmin -getServiceState nn1
standby
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$
```

將nn1切斷後，需重新啟動才能回復  
為standby狀態

Overview 'bdse33.example.org:8020' (active)

Namespace:	nncluster
NameNode ID:	nn1
Started:	Sun Jul 11 11:21:58 +0800 2021
Version:	3.3.0, raa96f1871bf9b58f9bac59c2a81ec470da649af
Compiled:	Tue Jul 07 02:44:00 +0800 2020 by brahma from branch-3.3.0
Cluster ID:	CID-62179020-e8ae-4d18-917c-88eb920f441
Block Pool ID:	BP-559628005-10.120.31.33-1623052072220

Summary

Security is off.  
Safemode is off.

Overview 'bdse138.example.org:8020' (ostandby)

Namespace:	nncluster
NameNode ID:	nn2
Started:	Sun Jul 11 11:21:58 +0800 2021
Version:	3.3.0, raa96f1871bf9b58f9bac59c2a81ec470da649af
Compiled:	Tue Jul 07 02:44:00 +0800 2020 by brahma from branch-3.3.0
Cluster ID:	CID-62179020-e8ae-4d18-917c-88eb920f441
Block Pool ID:	BP-559628005-10.120.31.33-1623052072220

Summary

Security is off.  
Safemode is off.

# RM HA

```
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ yarn rmadmin -getServiceState rm1
active
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ yarn rmadmin -getServiceState rm2
standby
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ yarn rmadmin -getServiceState rm1
^[[O2021-07-11 14:30:28,791 INFO ipc.Client: Retrying connect to server: bdse113.example.org/10.120.31.113:8033. Already tried 0 time(s); retry policy is Reti
UpToMaximumCountWithFixedSleep(maxRetries=1, sleepTime=1000 MILLISECONDS)
Operation failed: Call From bdse34.example.org/10.120.31.34 to bdse113.example.org:8033 failed on connection exception: java.net.ConnectException: Connection
Refused; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ yarn rmadmin -getServiceState rm1
2021-07-11 14:30:39,224 INFO ipc.Client: Retrying connect to server: bdse113.example.org/10.120.31.113:8033. Already tried 0 time(s); retry policy is RetryUp
MaximumCountWithFixedSleep(maxRetries=1, sleepTime=1000 MILLISECONDS)
Operation failed: Call From bdse34.example.org/10.120.31.34 to bdse113.example.org:8033 failed on connection exception: java.net.ConnectException: Connection
Refused; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$ yarn rmadmin -getServiceState rm2
active
hadoop@bdse34:/usr/local/hadoop/etc/hadoop$
```



(18008372筆資料)

## Start PySpark Session

```
[4]: # only for standalone pyspark, Light pyspark, if not native, appname = data_processing
# create spark session object (Local mode)
spark=SparkSession.builder.appName('data_processing').getOrCreate()
```

```
[5]: # Check spark app name
      spark.sparkContext.appName
```

```
[5]: 'PySparkShell'
```

```
[6]: spark
```

[6]: SparkSession - hive

## SparkContext

Spark III

```
Version      v3.1.2
Master       yarn
AppName      PySparkShell
```

```
[7]: spark.conf.set("spark.sql.execution.arrow.pyspark.enabled", True)
```

```
[8]: # need export ARROW_PRE_0_15_IPC_FORMAT=1 in .bashrc
import databricks.koalas as ks
```

```
[9]: ks.set_option("compute.default_index_type", "distributed")
```

### Load performance data

```
[10]: %%timeit
# Read CSV file
df_train = ks.read_csv('/user/spark/share/performance_detail.csv')
20.9 s ± 4.94 s per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
[11]: %timeit
# Read CSV file
dfs_train = spark.read.csv('/user/spark/share/performance_detail.csv', inferSchema=True, header=True)

12.8 s ± 322 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

檔案與其他

## II Applications

### Load performance data

```
[10] %%timeit
# Read CVS file
df_train = ks.read_csv('/user/spark/share/performance_detail.csv')
20.9 s ± 4.94 s per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
[11]: %timeit
# Read CVS file
dfs_train = spark.read.csv('/user/spark/share/performance_detail.csv', inferSchema=True, header=True)

12.8 s ± 322 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
[12]: df_train = k.read_csv('/user/spark/share/performance_detail.csv')
      dfs_train = spark.read_csv('/user/spark/share/performance_detail.csv')

[13]: %%timeit
      df_train.shape

2.9 s ± 1.1 s per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
[14]: df_train.shape
```

```
[15] %%timeit
(df_train.count(), len(df_train.columns))

2.44 s ± 180 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
[16]: (dfs_train.count(), len(dfs_train.columns))
```

```
[16]: (18008373, 110)
```

Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
36 GB	42 GB	0 B	6	12
Lost Nodes		Unhealthy Nodes		Rebooted Nodes

Cluster Metrics																			
12.8 s ± 322 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)																			
<div>1 Apps Submitted0 Apps Pending1 Apps Running0 Apps Completed6 Containers Running36 GB Memory Used42 GB Memory Total0 B Memory Reserved6 VCore Used12 VCore Total</div>																			
Cluster Nodes Metrics																			
<div>6 Active Nodes0 Decommissioning Nodes0 Decommissioned Nodes0 Lost Nodes0 Unhealthy Nodes0 Rebooted Nodes0</div>																			
Scheduler Metrics																			
<div>Scheduler TypeScheduling Resource TypeMinimum AllocationMaximum AllocationMaximum Cluster Ap</div>																			
<div>Capacity Scheduler[memory-mb (unit-M), vcores]&lt;memory:1024, vCores:1&gt;&lt;memory:7168, vCores:2&gt;0</div>																			
Show 20 entries																			
ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU V-Cores	Allocated Memory MB	Reserved CPU V-Cores	Reserved Memory MB	% of Queue	% of Cluster	Pro
application-16277068140001_0001	hadoop	PySparkShell	SPARK		default	0	Sat Jul 31 12:49:54 -0800 2021	Sat Jul 31 12:49:56 +0800 2021	N/A	RUNNING	UNDEFINED	6	6	36864	0	0	85.7	85.7	
application-1627694255516_0008	hadoop	PySparkShell	SPARK		default	0	Sat Jul 31 11:47:13 -0800 2021	Sat Jul 31 11:49:24 +0800 2021	Sat Jul 31 11:49:24 +0800 2021	FINISHED	FAILED	N/A	N/A	N/A	N/A	N/A	0.0	0.0	

# Troubleshooting

啟動 historyserver, historyserver 自己輸入jps做確認時發現螢幕未顯示出JobHistoryServer

- ◆ cd /usr/local/hadoop
- ◆ ls -l 有 logs
- ◆ cd /usr/local/hadoop/logs
- ◆ less -N 看 hadoop-hadoop-historyserver-bdsexx.logs 檔案

```
155 rver(JobHistoryServer.java:224)
156     at org.apache.hadoop.mapreduce.v2.hs.JobHistoryServer.main(JobHistorySer
156 ver.java:223)
157 Caused by: java.net.BindException: Port in use: master3.example.org:19888
158     at org.apache.hadoop.http.HttpServer2.constructBindException(HttpServer2
158 .java:1292)
159     at org.apache.hadoop.http.HttpServer2.bindForSinglePort(HttpServer2.jav
159 :1314)
160     at org.apache.hadoop.http.HttpServer2.openListeners(HttpServer2.java:137
160 3)
161     at org.apache.hadoop.http.HttpServer2.start(HttpServer2.java:1223)
162     at org.apache.hadoop.yarn.webapp.WebApps$Builder.start(WebApps.java:472)
163     at org.apache.hadoop.yarn.webapp.WebApps$Builder.start(WebApps.java:472)
```



# Troubleshooting

- ◆ `cd $HADOOP_HOME`
- ◆ `cd etc/hadoop`
- ◆ `nano mapred-site.xml`

Change `master3.example.org:19888` to  
`bdse57.example.org:19888`

<figure>

OKAY

```
    </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>bdse57.example.org:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>bdse57.example.org:19888</value>
  </property>
</configuration>
```

# 測試執行程式

```
[1]: import time

import numpy as np
import pandas as pd
import pyspark
import sys

[2]: import pyspark.sql.functions as fn

[3]: from pyspark.ml.regression import RandomForestRegressor
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.feature import VectorAssembler

[8]: import databricks.koalas as ks

[9]: df = ks.read_csv('/user/spark/share/AF_Stock_TW_2603.TW.csv')

[10]: df = df.drop('key_0', axis='columns')

[11]: tar = ks.DataFrame(df['close']).shift(periods=20).rename('label')

[12]: ks.set_option('compute.ops_on_diff_frames', True)
df = ks.concat([df, tar], axis=1)

[13]: # 保留前 100 row 空值小於等於 60% 的 column
cond = df.iloc[:101,:].isnull().sum()/100 <= 0.6
df = df[cond[cond == True].index.to_numpy()]

[14]: # 看 58 row 以後的空值
df.iloc[58,:].isnull().any().sum()

[14]: 0

[15]: # 取 58 row 以後重新 df # index reset
df = df.iloc[58,:].reset_index(drop=True)

[16]: df.shape

[16]: (2508, 184)

[17]: sdf = df.to_spark()
```

## Build Random Forest Model

```
[25]: rf = RandomForestRegressor()
rf_model = rf.fit(train_df)

[26]: # predict on the test set
model_predictions = rf_model.transform(test_df)

[27]: # print prediction
model_predictions.show(10)
```

features	label	prediction
[19.2769391502943...	19.276939392089844	19.18613926767244
[19.6624788274294...	18.660076141357425	19.10773054039324
[19.8937997357087...	19.276939392089844	19.18613926767244
[19.8937998275340...	17.619121551513672	19.019794481807274
[20.4335541213312...	18.852846145629883	19.72018766766709
[21.2046327145946...	19.778139114379883	20.249421125173686
[21.2817385819022...	19.084169387817383	19.78803131285374
[21.5901708137132...	19.35404586791992	20.057975469364557
[183,[0,1,2,3,4,5...	19.97090721130371	20.249421125173686
[19.4311537671171...	20.317893981933597	20.30185747590754

only showing top 10 rows

## Evaluate Model

```
[28]: evaluatorRMSE = RegressionEvaluator().setLabelCol('label').setPredictionCol("prediction").setMetricName("rmse")
```

```
[29]: RMSE = evaluatorRMSE.evaluate(model_predictions)
RMSE
```

```
[29]: 1.282201095799959
```

```
[30]: evaluatorR2 = RegressionEvaluator().setLabelCol('label').setPredictionCol("prediction").setMetricName("r2")
```

```
48]: # FEATURE IMPORTANCES
```

```
best_rf_model.featureImportances
```

```
48]: SparseVector(183, {0: 0.0, 1: 0.0001, 2: 0.0, 3: 0.0, 4: 0.0003, 5: 0.0043, 6: 0.0001, 7: 0.0007, 8: 0.0, 9: 0.0, 10: 0.0, 11: 0.0, 12: 0.0, 13: 0.0, 14: 0.0, 15: 0.0, 16: 0.0, 17: 0.0, 18: 0.0, 19: 0.0, 20: 0.0, 21: 0.015, 22: 0.0, 23: 0.0, 24: 0.0011, 25: 0.0, 26: 0.0, 27: 0.0049, 28: 0.0, 29: 0.0, 30: 0.0003, 31: 0.0003, 32: 0.0, 33: 0.0, 34: 0.0, 35: 0.0, 36: 0.0, 37: 0.0, 38: 0.0, 39: 0.0, 40: 0.0, 41: 0.0, 42: 0.0, 43: 0.0, 44: 0.0001, 45: 0.0001, 46: 0.0001, 47: 0.0, 48: 0.0007, 49: 0.0002, 50: 0.0001, 51: 0.0002, 52: 0.0002, 53: 0.0, 54: 0.0, 55: 0.0, 56: 0.0, 57: 0.0, 58: 0.0, 59: 0.0, 60: 0.0, 61: 0.0, 62: 0.0, 63: 0.0, 64: 0.0, 65: 0.0013, 66: 0.0001, 67: 0.0002, 68: 0.0, 69: 0.0, 70: 0.0, 71: 0.0007, 72: 0.0001, 73: 0.0, 74: 0.0, 75: 0.0, 76: 0.0, 77: 0.0, 78: 0.0, 79: 0.0, 80: 0.0, 81: 0.0, 82: 0.0007, 83: 0.0, 84: 0.0, 85: 0.0001, 86: 0.0003, 87: 0.1565, 88: 0.0, 89: 0.045, 90: 0.0, 91: 0.0, 92: 0.0, 93: 0.0, 94: 0.0, 95: 0.0, 96: 0.0, 97: 0.0, 98: 0.0, 99: 0.0, 100: 0.0, 101: 0.0, 102: 0.0, 103: 0.0, 104: 0.0, 105: 0.0, 106: 0.0, 107: 0.0, 108: 0.0, 109: 0.0, 110: 0.0, 111: 0.0, 112: 0.0, 113: 0.0, 114: 0.0, 115: 0.0, 116: 0.0, 117: 0.0, 118: 0.0, 119: 0.0, 120: 0.0, 121: 0.0, 122: 0.0, 123: 0.0, 124: 0.0, 125: 0.0, 126: 0.0, 127: 0.0, 128: 0.0, 129: 0.0, 130: 0.0, 131: 0.0, 132: 0.0, 133: 0.0, 134: 0.0, 135: 0.0, 136: 0.0, 137: 0.0, 138: 0.0, 139: 0.0, 140: 0.0, 141: 0.0, 142: 0.0, 143: 0.0, 144: 0.0, 145: 0.0, 146: 0.0, 147: 0.0, 148: 0.0, 149: 0.0, 150: 0.0, 151: 0.0, 152: 0.0, 153: 0.0, 154: 0.0, 155: 0.0, 156: 0.0014, 157: 0.0003, 158: 0.0, 159: 0.0001, 160: 0.0, 161: 0.0, 162: 0.0, 163: 0.0, 164: 0.0, 165: 0.0, 166: 0.0, 167: 0.0, 168: 0.0, 169: 0.0, 170: 0.0, 171: 0.0, 172: 0.0, 173: 0.0024, 174: 0.0735, 175: 0.0268, 176: 0.0, 177: 0.0001, 178: 0.0001, 179: 0.003, 180: 0.0, 181: 0.0, 182: 0.0, 183: 0.0})
```



THANKS FOR YOUR ATTENTION

---