EXECUTION & OUTPUTS

```
user@user-Latitude-3400:~$ ssh cs3304.057@abacus.iiit.ac.in
cs3304.057@abacus.iiit.ac.in's password:
```

First Time:

```
[cs3304.057@abacus ~]$ mkdir 2023201058_Assignment5
[cs3304.057@abacus ~]$ cd 2023201058_Assignment5/
[cs3304.057@abacus 2023201058_Assignment5]$ python3 -m venv
.aos
[cs3304.057@abacus 2023201058_Assignment5]$ source
.aos/bin/activate
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ pip install
wheel; pip install pyspark
```

Move csv and python files into **2023201058_Assignment5** folder and execute python files

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ deactivate;
exit
```

Every Time:

```
[cs3304.057@abacus ~]$ cd 2023201058_Assignment5/; source
.aos/bin/activate
```

Start python file executions

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ deactivate;
exit
```

- **Q1 #cores = 2**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 2
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q1.py ; exit ;
```

OUTPUT:

```
Second Highest Value Transaction among Selected Countries :
        Transaction unique identifier :
{2A289EA0-C13B-CDC8-E050-A8C063054829}
        Price :  20000000
        Country :  GREATER LONDON
Time taken for execution in seconds : 42.57427382469177
```

- **Q1 #cores = 4**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 4
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q1.py ; exit ;
```

OUTPUT:

```
Second Highest Value Transaction among Selected Countries :
        Transaction unique identifier :
{2A289EA0-C13B-CDC8-E050-A8C063054829}
        Price :  20000000
        Country :  GREATER LONDON
Time taken for execution in seconds : 27.33772110939026
```

- **Q1 #cores = 6**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 6
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q1.py ; exit ;
```

OUTPUT:

Second Highest Value Transaction among Selected Countries :
_____Transaction unique identifier :
{2A289EA0-C13B-CDC8-E050-A8C063054829}
_____Price : 20000000
_____Country : GREATER LONDON
Time taken for execution in seconds : 22.457929849624634

- **Q1 #cores = 6**

(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ **sint3**
[cs3304.057@node0x 2023201058_Assignment5]$ **python3 2023201058_q1.py ; exit ;**

OUTPUT:

Second Highest Value Transaction among Selected Countries :
_____Transaction unique identifier :
{2A289EA0-C13B-CDC8-E050-A8C063054829}
_____Price : 20000000
_____Country : GREATER LONDON
Time taken for execution in seconds : 22.049203634262085

- **Q1 #cores = 24**

(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ **python3 2023201058_q1.py**

OUTPUT:

Second Highest Value Transaction among Selected Countries :
_____Transaction unique identifier :
{2A289EA0-C13B-CDC8-E050-A8C063054829}
_____Price : 20000000
_____Country : GREATER LONDON
Time taken for execution in seconds : 14.350224494934082

- **Q2 #cores = 2**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 2
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q2.py ; exit ;
```

OUTPUT:

```
The country with the second most/highest transactions : GREATER MANCHESTER
Number of transactions in GREATER MANCHESTER country : 198338
Time taken for execution in seconds : 43.63918471336365
```

- **Q2 #cores = 4**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 4
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q2.py ; exit ;
```

OUTPUT:

```
The country with the second most/highest transactions : GREATER MANCHESTER
Number of transactions in GREATER MANCHESTER country : 198338
Time taken for execution in seconds : 27.88830304145813
```

- **Q2 #cores = 6**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 6
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q2.py ; exit ;
```

OUTPUT:

```
The country with the second most/highest transactions : GREATER MANCHESTER
Number of transactions in GREATER MANCHESTER country : 198338
Time taken for execution in seconds : 22.318350076675415
```

- **Q2 #cores = 6**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q2.py ; exit ;
```

OUTPUT:

```
The country with the second most/highest transactions : GREATER MANCHESTER
Number of transactions in GREATER MANCHESTER country : 198338
Time taken for execution in seconds : 22.029314279556274
```

- **Q2 #cores = 24**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ python3
2023201058_q2.py
```

OUTPUT:

```
The country with the second most/highest transactions : GREATER MANCHESTER
Number of transactions in GREATER MANCHESTER country : 198338
Time taken for execution in seconds : 15.138737916946411
```

- **Q3 #cores = 2**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 2
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q3.py ; exit ;
```

OUTPUT:

A CSV file named Country_Transaction_Counts_q3.csv is created in 2023201058_q3 folder.
The above CSV file contains Number of Transactions per Country
Time taken for execution in seconds : 37.46738910675049

- **Q3 #cores = 4**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 4
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q3.py ; exit ;
```

OUTPUT:

A CSV file named Country_Transaction_Counts_q3.csv is created in 2023201058_q3 folder.
The above CSV file contains Number of Transactions per Country
Time taken for execution in seconds : 25.784955739974976

- **Q3 #cores = 6**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3 -c 6
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q3.py ; exit ;
```

OUTPUT:

A CSV file named Country_Transaction_Counts_q3.csv is created in 2023201058_q3 folder.
The above CSV file contains Number of Transactions per Country
Time taken for execution in seconds : 21.745280981063843

- **Q3 #cores = 6**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ sint3
[cs3304.057@node0x 2023201058_Assignment5]$ python3
2023201058_q3.py ; exit ;
```

OUTPUT:

A CSV file named Country_Transaction_Counts_q3.csv is created in 2023201058_q3 folder.
The above CSV file contains Number of Transactions per Country
Time taken for execution in seconds : 21.13943099975586

- **Q3 #cores = 24**

```
(.aos) [cs3304.057@abacus 2023201058_Assignment5]$ python3
2023201058_q3.py
```

OUTPUT:

A CSV file named Country_Transaction_Counts_q3.csv is created in 2023201058_q3 folder.
The above CSV file contains Number of Transactions per Country
Time taken for execution in seconds : 15.141966104507446

Common Warning in All of the Executions:

```
Using Spark's default log4j profile:
org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For
SparkR, use setLogLevel(newLevel).
```

```
yy/mm/dd hh:mm:ss WARN NativeCodeLoader: Unable to load
native-hadoop library for your platform... using builtin-java
classes where applicable
/home/iiit/cs3304.057/2023201058_Assignment5/.aos/lib64/python3
.6/site-packages/pyspark/context.py:238: FutureWarning: Python
3.6 support is deprecated in Spark 3.2.
 FutureWarning
```

OBSERVATIONS

Execution Time (sec)	Q1	Q2	Q3
Sint3 -c 2 #cores = 2	**42.57427382**	**43.63918471**	**37.46738911**
%age decrease	35.78816817	36.09343707	31.18027075
Sint3 -c 4 #cores = 4	**27.33772111**	**27.88830304**	**25.78495574**
%age decrease	17.85002942	19.97236245	15.66678958
Sint3 -c 6 #cores = 6	**22.45792985**	**22.31835008**	**21.74528098**
%age decrease	1.819963897	1.295058981	2.78612165
Sint3 #cores = 6	**22.04920363**	**22.02931428**	**21.139431**
%age decrease	34.91726625	31.27912324	33.10148175
Direct Execution #cores = 24	**14.35022449**	**15.13873792**	**14.1419661**

CONCLUSIONS

1. Generally, increasing the number of cores leads to a decrease in execution time, indicating improved parallel processing efficiency.
2. However, there is diminishing returns as the number of cores increases, especially evident when moving from 4 to 6 cores.
3. There is a noticeable decrease in execution time when moving from 2 to 4 cores.
4. The decrease in execution time from 4 to 6 cores is still observed but is less significant.
5. Moving from 6 cores to direct execution with 24 cores shows a substantial decrease in execution time.

The percentage decrease in execution time is not proportionally the same as the number of cores increases because the relationship between the number of cores and parallel processing efficiency is not linear. The efficiency gains from parallelization often exhibit diminishing returns.

Here are a few reasons why the percentage decrease may not be proportionally the same:

1. **Parallelization Overhead:**
    - As you increase the number of cores, there might be additional overhead associated with managing the parallel processes. This overhead can reduce the overall speedup gained from parallelization.
2. **Workload Characteristics:**
    - The nature of the workload and how well it can be parallelized play a significant role. Some tasks are more easily parallelizable, while others may have dependencies or limitations that prevent efficient parallelization.
3. **Amdahl's Law:**
    - Amdahl's Law states that the speedup of a program using multiple processors is limited by the sequential portion of the program. If there are parts of the program that cannot be parallelized, the overall speedup will be limited.
4. **Communication Overhead:**
    - In some parallel processing scenarios, the communication between cores or nodes can introduce overhead. As the number of cores increases, the impact of communication overhead may become more significant.
5. **Resource Contention:**
    - Increasing the number of cores may lead to increased contention for shared resources (e.g., memory, I/O). This contention can introduce

bottlenecks and limit the scalability of performance improvement.

6. **Algorithmic Efficiency:**
    - The efficiency of the parallel algorithm used in the program can also impact the scalability. Some algorithms scale well with increased parallelism, while others may have limitations.

In summary, the relationship between the number of cores and performance improvement is complex and depends on various factors. While adding more cores generally improves parallel processing efficiency, the actual speedup is influenced by factors such as parallelization overhead, workload characteristics, and the efficiency of the parallel algorithm. This complexity leads to non-linear trends in percentage decrease as the number of cores increases.