ANALYZING LARGE-SCALE TRANSACTIONAL DATA: A DISTRIBUTED FP-GROWTH IMPLEMENTATION ON SPARK

GROUP MEMBERS

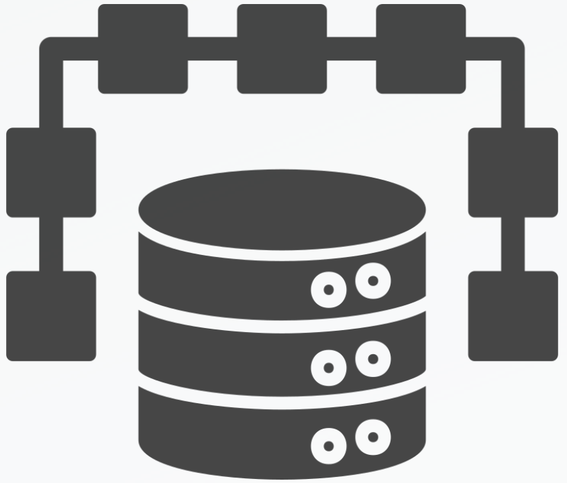
Bharath L	(22bds013)
Gnanesh A R	(22bds023)
Gopal	(22bds025)
Madhan S	(22bds036)
Suhaas	(22bds056)

CONTENT

HADOOP

What is Hadoop?

Hadoop is an open-source framework used for storing and processing large volumes of data in a distributed computing environment. It's designed to handle massive datasets across clusters of commodity hardware.



Key Components:

1. Hadoop Distributed File System (HDFS): A distributed file system that stores data across multiple machines.
2. MapReduce: A programming model for processing and generating large datasets in parallel.
3. YARN (Yet Another Resource Negotiator): Resource management layer responsible for managing computing resources in the Hadoop ecosystem.
4. Hadoop Common: Contains libraries and utilities needed by other Hadoop modules.

SPARK

- Open-source, in-memory application framework.

- Provides parallel & distributed processing, scalability & fault tolerance.

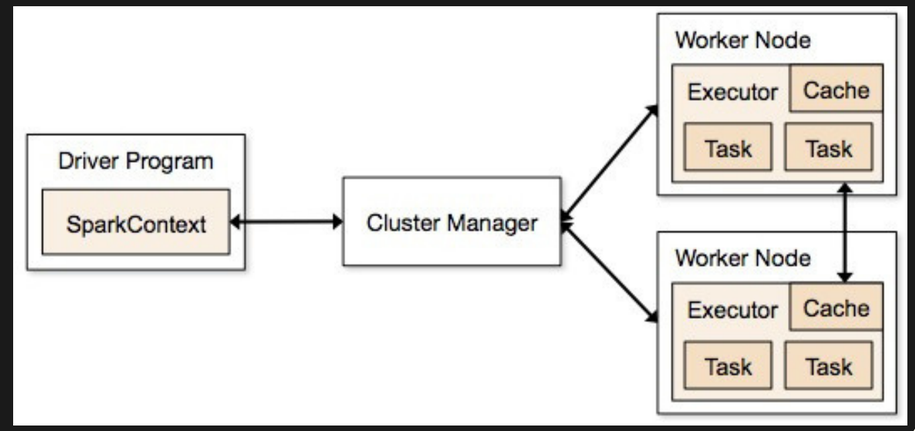
- Keeps as much of the data required in-memory & avoids disk I/O.

- Follows a master/slave architecture with two main daemons and a cluster manager (Master Daemon & Worker Daemon)

Apache Spark is a distributed computing framework that provides in-memory data processing and processing of batch, streaming, and machine learning workloads. It can be used with various data sources such as HDFS, HBase, Cassandra, and Amazon S3 and supports multiple programming languages including Java, Scala, and Python.

MASTER & WORKER DAEMON

- Spark consists of the driver program & the executor program.
- Executor program works on worker nodes.
- Spark distributes RDDs among executors.
- Communication among the driver & the executor.
- Spark as an organization :-



1. Executive management as driver code.
2. Junior employees as executors.
3. Worker nodes correspond to the office space that the employees occupy.

RDD'S

• Spark's primary data abstraction.

• Collection of fault tolerant & immutable elements

partitioned across the cluster's nodes. • Spark create a DAG when creating an RDD.

• Vertices represent the RDD's & the edges represent the transformations or actions.

• The DAG scheduler applies the graphical structure to run the tasks that use the RDD to perform transformational processes.

• DAG helps enable fault tolerance. When a node goes down, Spark replicates the DAG & restores the node

FP - GROWTH ALGORITHM

“FP” stands for Frequent Pattern in a Dataset of transactions

1. calculate item frequencies and identify frequent items,
2. a suffix tree (FP-tree) structure to encode transactions, and
3. frequent itemsets can be extracted from the FP-tree.

Frequency of each item

TID	Items Bought
100	*f, a, c, d, g, i, m, p*
200	*a, b, c, f, l, m, o*
300	*b, f, h, j, o*
400	*b, c, k, s, p*
500	*a, f, c, e, l, p, m, n*

Frequency of each item

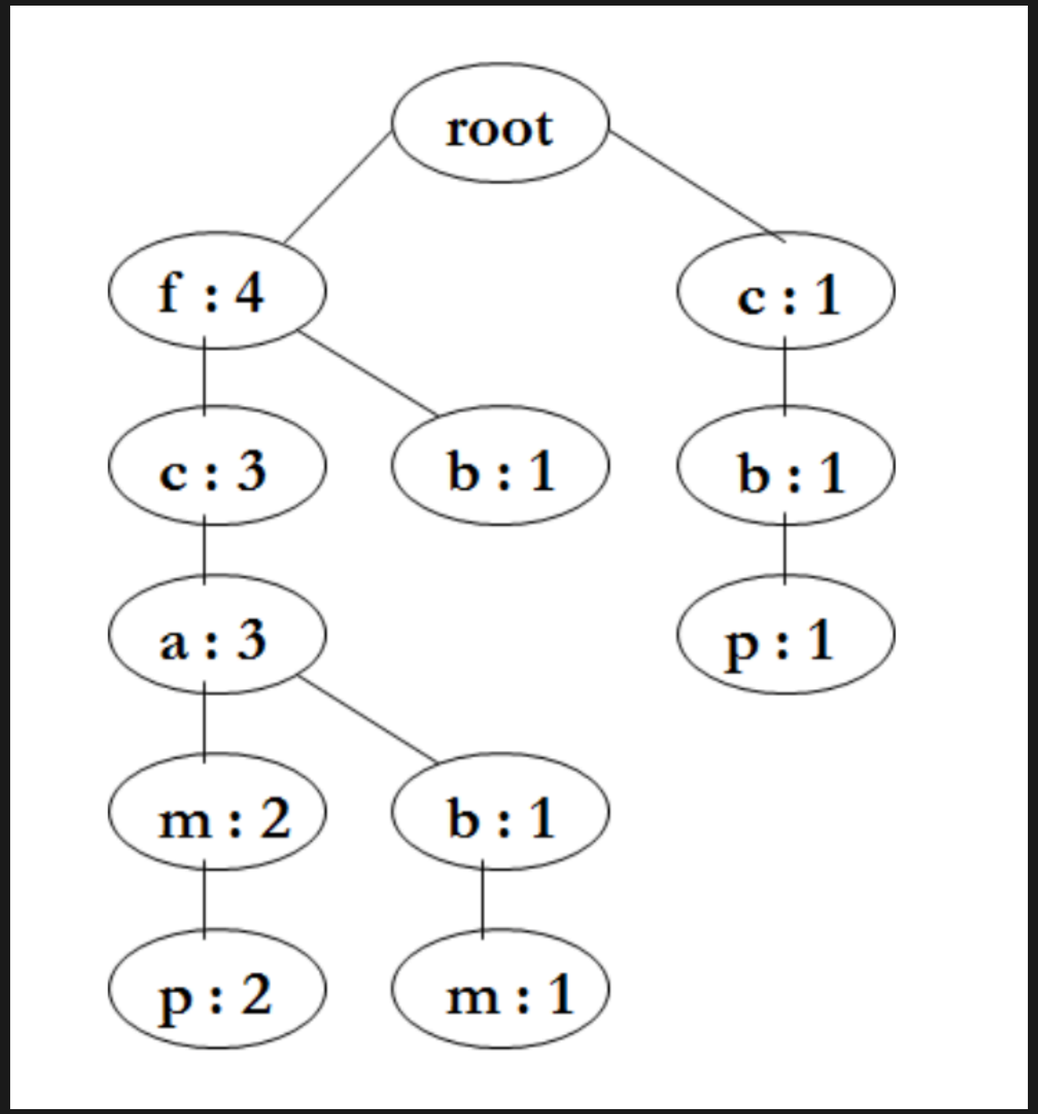
Item	Frequency	Item	Frequency
a	3	j	1
b	3	k	1
c	4	l	2
d	1	m	3
e	1	n	1
f	4	o	2
g	1	p	3
h	1	s	1
i	1		

Frequent Pattern set L = { (f:4), (c:4), (a:3), (b:3), (m:3), (p:3) }

Ordered Item set

TID	Items Bought	(Ordered) Frequent Items
100	*f, a, c, d, g, i, m, p*	*f, c, a, m, p*
200	*a, b, c, f, l, m, o*	*f, c, a, b, m*
300	*b, f, h, j, o*	*f, b*
400	*b, c, k, s, p*	*c, b, p*
500	*a, f, c, e, l, p, m, n*	*f, c, a, m, p*

Frequent Pattern Tree



Conditional Pattern Base

Item	Conditional Pattern Base
p	{{f, c, a, m : 2}, {c, b : 1}}
m	{{f, c, a : 2}, {f, c, a, b : 1}}
b	{{f, c, a : 1}, {f : 1}, {c : 1}}
a	{{f, c : 3}}
c	{{f : 3}}
f	Φ

Conditional Frequent Pattern - Tree

Item	Conditional Pattern Base	Conditional FP-Tree
p	{{f, c, a, m : 2}, {c, b : 1}}	{c : 3}
m	{{f, c, a : 2}, {f, c, a, b : 1}}	{f, c, a :3}
b	{{f, c, a : 1}, {f : 1}, {c : 1}}	Φ
a	{{f, c : 3}}	{f, c : 3}
c	{{f : 3}}	{f : 3}
f	Φ	Φ

Frequent Patterns Generated

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Patterns Generated
p	{{f, c, a, m : 2}, {c, b : 1}}	{c : 3}	{<c, p : 3>}
m	{{f, c, a : 2}, {f, c, a, b : 1}}	{f, c, a :3}	{ <f, m : 3>, <c, m : 3> <a, m : 3>, <f, c, m : 3> <f, a, m : 3>, <c, a, m :3>}
b	{{f, c, a : 1}, {f : 1}, {c : 1}}	Φ	{}
a	{{f, c : 3}}	{f, c : 3}	{<f, a : 3>, <c, a : 3>, <f, c, a:3>}
c	{{f : 3}}	{f : 3}	{ <f, c : 3>}
f	Φ	Φ	{}

CODE IMPLEMENTATION

```
+---+--------------------+--------------------+
|  0|[bread, jelly, pe...|[jam, milk, chees...|
|  1|[bread, peanut_bu...|[jam, milk, chees...|
|  2|[bread, milk, pea...|[jam, butter, bee...|
|  3|       [beer, bread]|[milk, cheese, ja...|
|  4|        [beer, milk]|[jam, butter, bre...|
|  5|      [bread, jelly]|[jam, milk, chees...|
|  6|     [bread, cheese]|[jam, milk, butte...|
|  7|     [bread, butter]|[jam, milk, chees...|
|  8|        [bread, jam]|[butter, milk, ch...|
|  9|  [bread, milk, jam]|[butter, cheese, ...|
| 10|[bread, butter, jam]|[milk, cheese, beer]|
| 11|[bread, butter, c...|   [jam, milk, beer]|
| 12|[bread, butter, m...|[jam, beer, biscu...|
| 13|[bread, milk, che...|[jam, butter, bee...|
| 14|[bread, butter, j...|[cheese, beer, bi...|
| 15|[bread, butter, j...|       [milk, beer]|
| 16|[bread, jam, milk...|[butter, beer, bi...|
| 17|[bread, butter, j...|[beer, biscuit, c...|
| 18|[beer, bread, pea...|[milk, cheese, ja...|
| 19|[beer, bread, jelly]|[milk, cheese, ja...|
+---+--------------------+--------------------+
only showing top 20 rows

--------------------------------------------------
Best hyperparameters:
MinSupport: 0.1
MinConfidence: 0.3
--------------------------------------------------
Total Time Taken (in Mins) :  0.2135839064915975
--------------------------------------------------
Node information saved to: /home/cluster/FP_Growth_Cluster_Node_Details.txt
24/04/28 02:45:41 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_42_piece0 on MADHANS:42021 in memory (size: 14.8 KiB, free: 366.2 MiB)
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_42_piece0 on 10.0.14.202:42975 in memory (size: 14.8 KiB, free: 366.2 MiB)
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_41_piece0 on MADHANS:42021 in memory (size: 14.8 KiB, free: 366.2 MiB)
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_41_piece0 on 10.0.14.202:42975 in memory (size: 14.8 KiB, free: 366.2 MiB)
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_44_piece0 on MADHANS:42021 in memory (size: 16.2 KiB, free: 366.3 MiB)
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_44_piece0 on 10.0.14.202:42975 in memory (size: 16.2 KiB, free: 366.3 MiB)
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_43_piece0 on MADHANS:42021 in memory (size: 12.9 KiB, free: 366.3 MiB)
24/04/28 02:45:41 INFO SparkUI: Stopped Spark web UI at http://MADHANS:4040
24/04/28 02:45:41 INFO BlockManagerInfo: Removed broadcast_43_piece0 on 10.0.14.202:42975 in memory (size: 12.9 KiB, free: 366.3 MiB)
24/04/28 02:45:41 INFO StandaloneSchedulerBackend: Shutting down all executors
24/04/28 02:45:41 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Asking each executor to shut down
24/04/28 02:45:41 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/04/28 02:45:41 INFO MemoryStore: MemoryStore cleared
24/04/28 02:45:41 INFO BlockManager: BlockManager stopped
24/04/28 02:45:41 INFO BlockManagerMaster: BlockManagerMaster stopped
24/04/28 02:45:41 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/04/28 02:45:41 INFO SparkContext: Successfully stopped SparkContext
24/04/28 02:45:42 INFO ShutdownHookManager: Shutdown hook called
24/04/28 02:45:42 INFO ShutdownHookManager: Deleting directory /tmp/spark-2fa4eda1-7a4e-48ec-96d4-4d49062b3eff
24/04/28 02:45:42 INFO ShutdownHookManager: Deleting directory /tmp/spark-6d8de26a-6599-4445-8c9c-67dd559e0896
24/04/28 02:45:42 INFO ShutdownHookManager: Deleting directory /tmp/spark-2fa4eda1-7a4e-48ec-96d4-4d49062b3eff/pyspark-1ba43598-fa3b-40b3-b555-b7c7b40f8dc1
cluster@MADHANS:~/spark-3.5.1$
```

* 1 Node

cluster@MADHANS: ~/spark-3.5.1

cluster@MADHANS: ~/spark-3.5.1 | cluster@MADHANS: ~

```
24/04/28 02:58:13 INFO DAGScheduler: Job 32 is finished. Cancelling potential speculative or zombie tasks for this job
24/04/28 02:58:13 INFO TaskSchedulerImpl: Killing all running tasks in stage 61: Stage finished
24/04/28 02:58:13 INFO DAGScheduler: Job 32 finished: showString at NativeMethodAccessorImpl.java:0, took 2.013450 s
24/04/28 02:58:13 INFO CodeGenerator: Code generated in 17.631055 ms
+---+--------------------+--------------------+
| id|               items|          prediction|
+---+--------------------+--------------------+
|  0|[bread, jelly, pe...|[cheese, butter, ...|
|  1|[bread, peanut_bu...|[cheese, butter, ...|
|  2|[bread, milk, pea...|[jam, butter, bee...|
|  3|       [beer, bread]|[jam, butter, mil...|
|  4|        [beer, milk]|[jam, butter, bread]|
|  5|      [bread, jelly]|[cheese, butter, ...|
|  6|     [bread, cheese]|[jam, butter, bee...|
|  7|     [bread, butter]|[cheese, beer, ja...|
|  8|         [bread, jam]|[cheese, beer, bu...|
|  9|  [bread, milk, jam]|[cheese, beer, bu...|
| 10|[bread, butter, jam]|[cheese, beer, milk]|
| 11|[bread, butter, c...|   [jam, beer, milk]|
| 12|[bread, butter, m...| [jam, beer, cheese]|
| 13|[bread, milk, che...| [jam, butter, beer]|
| 14|[bread, butter, j...|      [cheese, beer]|
| 15|[bread, butter, j...|        [beer, milk]|
| 16|[bread, jam, milk...|      [beer, butter]|
| 17|[bread, butter, j...|            [beer]|
| 18|[beer, bread, pea...|[jam, butter, mil...|
| 19|[beer, bread, jelly]|[jam, butter, mil...|
+---+--------------------+--------------------+
only showing top 20 rows

------------------------------------------------
Best hyperparameters:
MinSupport: 0.2
MinConfidence: 0.3
------------------------------------------------
Total Time Taken (in Mins) :  1.0883307576179504
------------------------------------------------
Node information saved to: /home/cluster/FP_Growth_Cluster_Node_Details.txt
24/04/28 02:58:13 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/04/28 02:58:13 INFO SparkUI: Stopped Spark web UI at http://MADHANS:4040
24/04/28 02:58:13 INFO StandaloneSchedulerBackend: Shutting down all executors
24/04/28 02:58:13 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Asking each executor to shut down
24/04/28 02:58:13 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/04/28 02:58:13 INFO MemoryStore: MemoryStore cleared
24/04/28 02:58:13 INFO BlockManager: BlockManager stopped
24/04/28 02:58:13 INFO BlockManagerMaster: BlockManagerMaster stopped
24/04/28 02:58:13 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/04/28 02:58:13 INFO SparkContext: Successfully stopped SparkContext
24/04/28 02:58:14 INFO ShutdownHookManager: Shutdown hook called
24/04/28 02:58:14 INFO ShutdownHookManager: Deleting directory /tmp/spark-f058af93-1979-43ca-a657-7fdad1654502
24/04/28 02:58:14 INFO ShutdownHookManager: Deleting directory /tmp/spark-f058af93-1979-43ca-a657-7fdad1654502/pyspark-fd88821a-415c-416d-bfe1-854f71e82b23
24/04/28 02:58:14 INFO ShutdownHookManager: Deleting directory /tmp/spark-b53162d3-f727-423d-9c73-1391e89b1468
cluster@MADHANS:~/spark-3.5.1$
```

* 3 Nodes

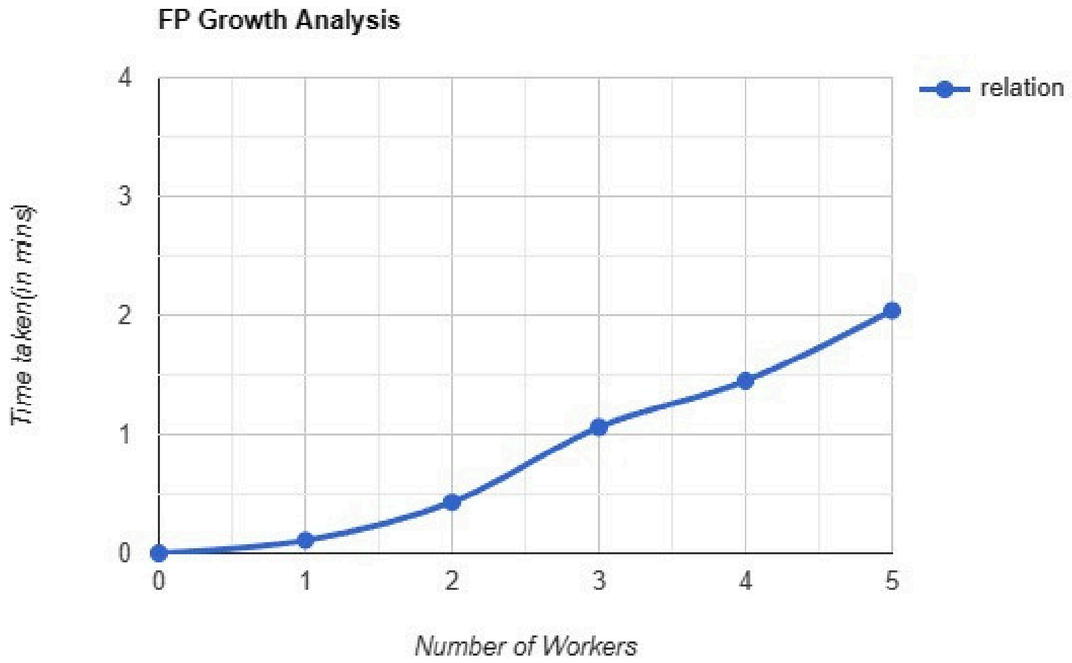
```
24/04/28 03:20:00 INFO DAGScheduler: Job 33 is finished. Cancelling potential speculative or zombie tasks for this job
24/04/28 03:20:00 INFO TaskSchedulerImpl: Killing all running tasks in stage 65: Stage finished
24/04/28 03:20:00 INFO DAGScheduler: Job 33 finished: showString at NativeMethodAccessorImpl.java:0, took 0.292914 s
24/04/28 03:20:00 INFO CodeGenerator: Code generated in 17.327917 ms
+---+--------------------+----------+
| id|               items|prediction|
+---+--------------------+----------+
|  0|[bread, jelly, pe...|        []|
|  1|[bread, peanut_bu...|        []|
|  2|[bread, milk, pea...|        []|
|  3|       [beer, bread]|        []|
|  4|        [beer, milk]|   [bread]|
|  5|      [bread, jelly]|        []|
|  6|     [bread, cheese]|        []|
|  7|     [bread, butter]|        []|
|  8|         [bread, jam]|        []|
|  9|  [bread, milk, jam]|        []|
| 10|[bread, butter, jam]|        []|
| 11|[bread, butter, c...|        []|
| 12|[bread, butter, m...|        []|
| 13|[bread, milk, che...|        []|
| 14|[bread, butter, j...|        []|
| 15|[bread, butter, j...|        []|
| 16|[bread, jam, milk...|        []|
| 17|[bread, butter, j...|        []|
| 18|[beer, bread, pea...|        []|
| 19|[beer, bread, jelly]|        []|
+---+--------------------+----------+
only showing top 20 rows

--------------------------------------------------
Best Hyperparameters:
MinSupport: 0.2
MinConfidence: 0.7
--------------------------------------------------
Total Time Taken (in Mins) :  2.060747762521108
--------------------------------------------------
Node Information saved to: /home/cluster/FP_Growth_Cluster_Node_Details.txt
24/04/28 03:20:01 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/04/28 03:20:01 INFO SparkUI: Stopped Spark web UI at http://MADHANS:4040
24/04/28 03:20:01 INFO StandaloneSchedulerBackend: Shutting down all executors
24/04/28 03:20:01 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Asking each executor to shut down
24/04/28 03:20:01 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/04/28 03:20:01 INFO MemoryStore: MemoryStore cleared
24/04/28 03:20:01 INFO BlockManager: BlockManager stopped
24/04/28 03:20:01 INFO BlockManagerMaster: BlockManagerMaster stopped
24/04/28 03:20:01 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/04/28 03:20:01 INFO SparkContext: Successfully stopped SparkContext
24/04/28 03:20:02 INFO ShutdownHookManager: Shutdown hook called
24/04/28 03:20:02 INFO ShutdownHookManager: Deleting directory /tmp/spark-2bdba80b-f64e-4152-8a5d-0dd0a394b6fe
24/04/28 03:20:02 INFO ShutdownHookManager: Deleting directory /tmp/spark-c79272ef-d7cb-444f-86b8-5f6b737ba9dc
24/04/28 03:20:02 INFO ShutdownHookManager: Deleting directory /tmp/spark-2bdba80b-f64e-4152-8a5d-0dd0a394b6fe/pyspark-b7e141af-7a57-4211-93f9-357ddf7f35f8
cluster@MADHANS:~/spark-3.5.1$
```

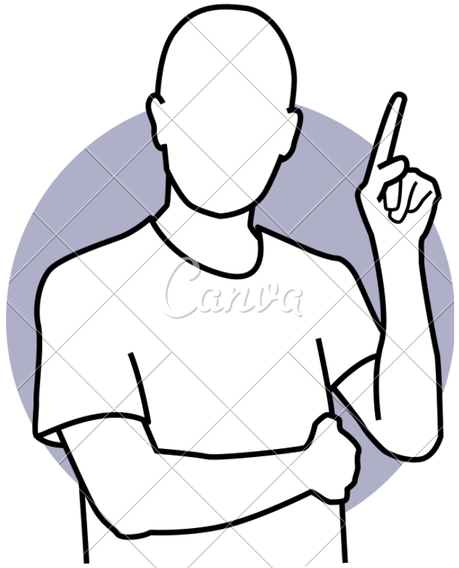
* 5 Nodes

ANALYSIS

The analysis of our project revealed an unexpected trend: as the number of nodes increased, the execution time of the FP-Growth algorithm also increased instead of decreasing.



REASON ?



NETWORK OVERHEAD 

Network overhead refers to the additional time, resources, and bandwidth consumed by communication between nodes in a distributed system.

This overhead can manifest in several ways:

1. Serialization and Deserialization
2. Network Latency
3. Data Shuffling
4. Protocol Overhead

UNDER THE GUIDANCE OF

Dr.Animesh Chaturvedi,
Associate Professor Dept. of DSAI , IIIT Dharwad.


THANK
YOU!