Parallel Programming & Performance on Clusters

Using the C Message Passing Interface

Design Document

Rachit Magon & Rishi Raj Sahu,

M.E. – Software Systems,

Bits Pilani, Pilani.

DESIGN OF THE MPI CLUSTER

In this section we have explained the design, implementation and assumptions taken with the MPI Cluster.

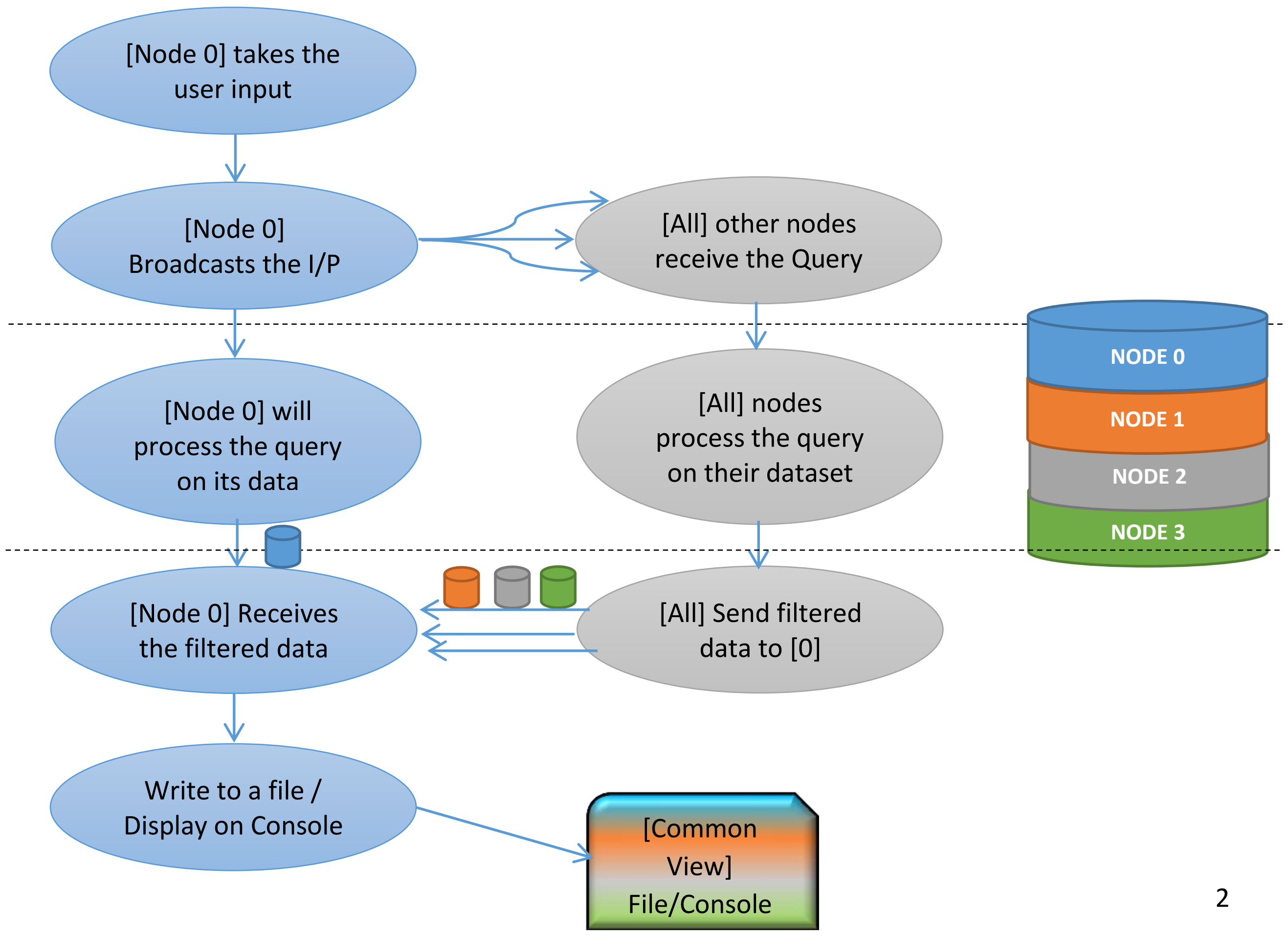
1.1 MPI – Partitioned Model

We have taken datasets (created using a script developed by us) of 5 different sizes: 10K, 100K, 1000K, 6000K and 10M rows and tested the data on 2,4,6 and 8 node clusters. The table was partitioned equally for all nodes, for example:

No. of nodes	No. of Records	No of Records/Node
2	10K	5K
4	10K	2.5K
8	10M	1250K

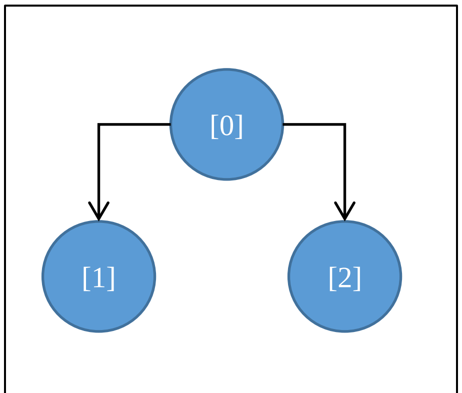
1.1.1 MPI – Partitioned Model – Select Query – Parallelization

Assuming a 4 node cluster with 10K records, the algorithm used for a simple Select Query follows:

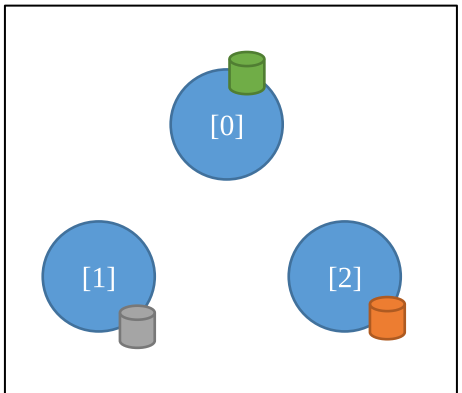

[Node 0] takes the user input
[Node 0] Broadcasts the I/P
[All] other nodes receive the Query
NODE 0
NODE 1
NODE 2
NODE 3
[Node 0] will process the query on its data
[All] nodes process the query on their dataset
[Node 0] Receives the filtered data
[All] Send filtered data to [0]
Write to a file / Display on Console
[Common View] File/Console

1.1.3 MPI – Partitioned Model – Join Query – Parallelization

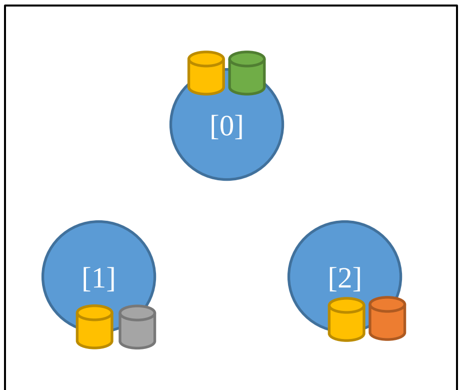
For simplicity, assuming a three node cluster with 10K records in total, the parallelization explanation follows:



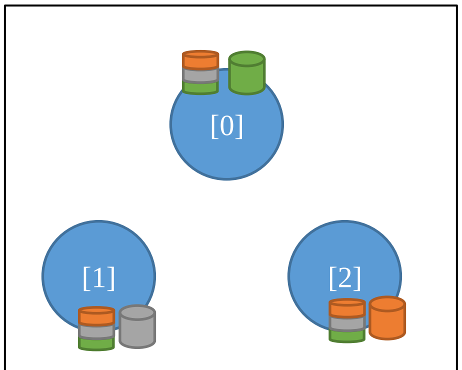
Step 1: Node [0] Broadcasts user Input to all other nodes



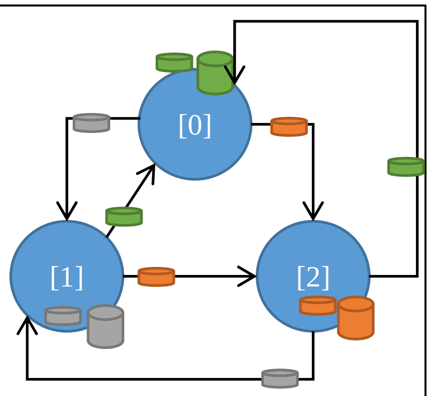
Step 2: Nodes get their primary tables from their respective



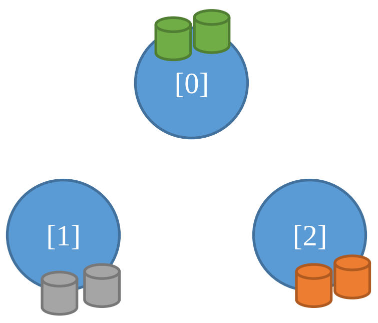
Step 3: Nodes get their secondary tables



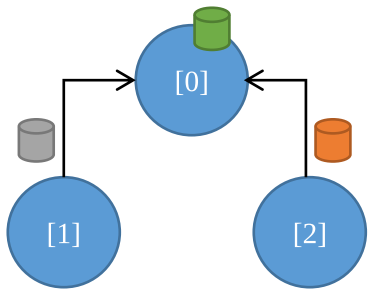
Step 4: Nodes partition the data in the secondary table according to the primary tables of others



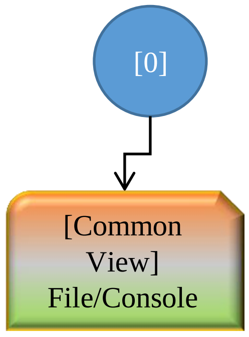
Step 5: Nodes send the data in accord to the partitions



Step 6: All nodes have consistent primary and secondary tables.



Step 7: All nodes send the joined data



Step 8: Node [0] displays the data to all nodes.

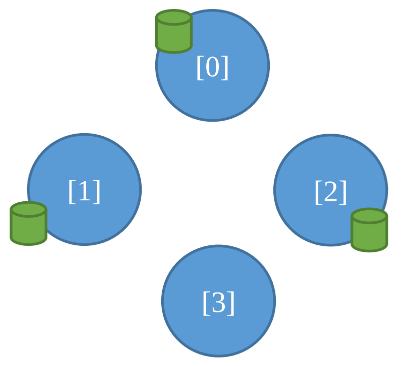
1.2 MPI – Replicated Model

We replicated the required tables in a few nodes but not all. The replication was increased as per the cluster size was increased:

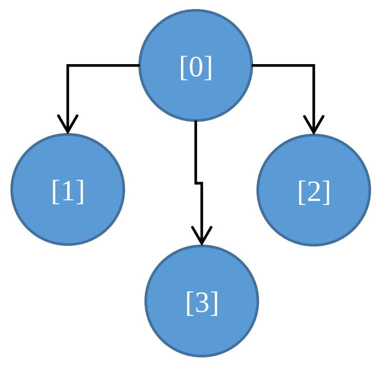
Cluster Size	Replication Factor
2	1
4	3
6	4
7	6

1.2.1 MPI – Replicated Model – Select Query – Parallelization

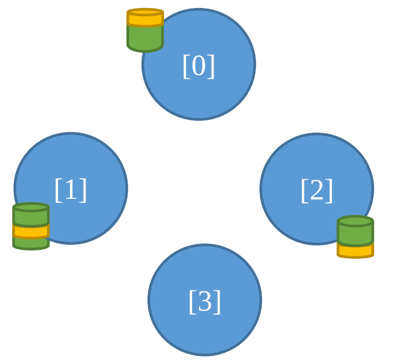
Assuming a four node cluster with the required table present in 3 nodes:



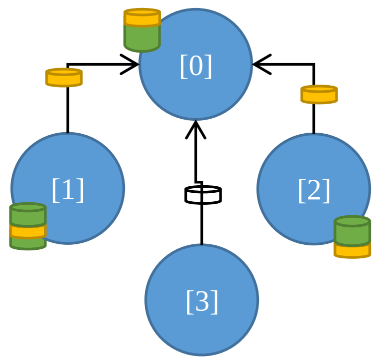
Step 1: All nodes get the partitioning information. NOTE: Node 3 does not have the table



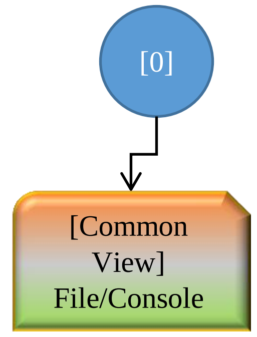
Step 2: Node [0] broadcasts the user I/P to all nodes



Step 3: Nodes calculate their share of the data to work on *Marked as Yellow*



Step 4: Nodes perform the select and send the data back to Node [0]



Step 5: Node [0] displays the data to all nodes.

1.2.3 MPI – Replicated Model – Join Query – Parallelism

The design idea behind the Join Query of Replicated model is as follows:

1. Node [0] gets the user input and distributes it to all nodes
2. All nodes parse the query and get the table names required to create the Join.
3. If a node does not have the tables required to do the join, it sends a 0 to the master Node [0].
4. If a node has the table required to do the join, it gets the Mapping and calculates the amount of data which can be processed by the node.
    a. For example, if there are 10,000 rows and there are 2 nodes which have the table required, each gets 5,000 nodes to process.
5. When the Join is calculated, the data is sent back to the master Node [0].
6. Node [0] prints the combined result.

Assumptions Made

General Assumptions & for Partitioned Model

1. Flat File tables have comma separated values.
2. Column names in the query are given along with table names i.e. SELECT TABLE.COLUMN FROM TABLE WHERE TABLE.COLUMN=5
3. There are no spaces in between a single condition i.e. TABLE.COLUMN=5 is correct but TABLE.COLUMN = 5 is wrong
4. Database files are present in the folder given by DBPATH, if DBPATH is null then the database folder resides with the program

5. METADATA file for the database is present at EVERY NODE. It should have a key,value pair <*,-1>.
6. Presently only one condition is supported after the WHERE clause.
7. There is a mapping file present for every table which tells the system about the partitioning. This is important for the JOIN operation to proceed successfully and will cause errors if not correct.
8. JOIN operation is on two tables only.
9. Query provided is correct.

Specifically for Replicated Model

1. Atleast one node should have both the tables required for JOIN.
2. Mapping file should be present with the code in a file named REPLICATED_MAPPING