Broadcast Join

What is a Broadcast Join?

- A join optimization technique in distributed data processing systems like Apache Spark, Hive, or Presto.
- Idea: If one table (dataset) is small enough to fit in memory, send (broadcast) it to all worker nodes instead of shuffling large datasets over the network.
- Each worker node then performs the join locally without needing more data transfer.

Why Broadcast Joins Exist

- In a **normal join**:
  - Both tables may be large.
  - Data is **shuffled** across the cluster so matching rows end up on the same worker node.
  - **Shuffling is expensive** — involves network I/O, disk writes, and sorting.
- In a **broadcast join**:
  - The small table is **replicated** to every worker node.
  - Only the big table is read locally → **avoids shuffling big data**.

When to Use

- One table is **small** (a few MBs or thousands of rows).
- Common in **dimension table + fact table** joins:
  - Example: sales (big) joined with country_codes (small).
- Small table fits into executor memory without causing out-of-memory errors.

4️⃣ How it Works

Normal Shuffle Join High network I/O.

Step 1: Partition both tables by join key

Step 2: Shuffle data across the network

Step 3: Join matching partitions

Broadcast Join

Step 1: Send small table to all worker nodes

Step 2: Each worker scans its part of the large table

Step 3: Joins with in-memory copy of the small table

No shuffle of large table.

6️⃣ Benefits

- **Performance**: Avoids shuffling large datasets → faster joins.
- **Resource efficiency**: Less network traffic and disk spill.
- **Good for star-schema joins** (fact + dimensions).

7 Trade-offs / Limitations

- The broadcasted table must **fit in memory** on every executor.
- Large broadcasts can cause **OutOfMemoryError**.
- Not beneficial if **both tables are large**.
- Each executor holds a full copy of the small table → **multiplies memory usage**.

8 Spark Configurations

- **Enable auto broadcast**:

sql

```
SET spark.sql.autoBroadcastJoinThreshold = 10MB; -- default
```

  - If the table size ≤ threshold → Spark auto-broadcasts it.
  - Set to -1 to disable auto-broadcast.

- **Force broadcast**:

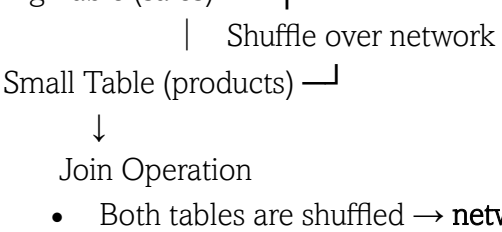
sql

```
SELECT /*+ BROADCAST(table_name) */ * FROM ...
```

  - Works even if table is larger than threshold (be careful with memory).

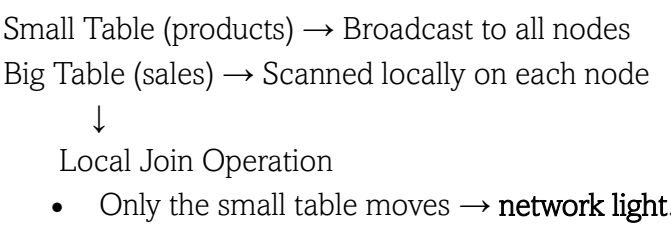
9 Before vs After Diagram

Normal Join (Shuffle)



- Both tables are shuffled → **network heavy**.

Broadcast Join



- Only the small table moves → **network light**.

---

10 Key Takeaways

- Use **broadcast join** when one table is **small** and fits in memory.
- Helps avoid expensive shuffle operations.
- Tune spark.sql.autoBroadcastJoinThreshold for your workload.
- Combine with **Dynamic Partition Pruning** for best performance in partitioned fact tables.
