

**The Justin Bieber Effect** - Hashing data to nodes is very good when the attribute chosen better approximates a uniform distribution. Certain nodes will become bottlenecks if a poorly chosen attribute is hashed.

**Partitioned Aggregation** - Hash shuffle tuples to correct nodes then aggregate locally.

**Implicit union** - Parallel query plans implicitly union (join all the tuples) at the end

**Partitioned Hash Equijoin Algorithm** (assume both tables are block partitioned)

1. Hash shuffle tuples on join attributes 2. Local join

**Hadoop**

**HadoopDFS (HDFS)**

-Stores and manages access to large files (tables)

-Files are partitioned into chunks (~64MB)

-Each chunk is replicated 3+ times to provide fault tolerance

**Hadoop MapReduce (easy to use distributed programming paradigm)**

-High disk usage

-Map (Format)

-Read data from disks

-Extract info from each tuple

-Transform it into a useful key-value format

-Shuffle (Group)

-key-value pairs into groups based on keys

-Reduce (Analyze)

-Perform analysis on groups

-Write results to disks

-MapReduce implements fault tolerance via writing results to disk (however write IOs are very slow )

**Spark**

-High memory usage

-Fast distributed processing on top of HDFS

-Compute everything in **memory** (RAM) but keep track of how it is computed

-Recovery from faults  is easy, fast, and efficient

-Resilient Distributed Dataset (RDD)

-A distributed, immutable relation and a lineage

-Java Objects:

* + SparkSession
  + Row
  + Dataset (Table) (SparkSQL)
  + JavaRDD<T> (Collection of type T) (Spark Functional API)

-The SparkSession object is an interface that lets us issue commands in Spark

-Make one for sparkCluster and sparkLocal

-SparkSQL queries must be optimized at runtime

-Java native Spark queries can be checked and optimized at compile time (better)

-Spark functional API:

-Transformations (map, reduce, join, filter, …)

* + Lazy evaluation
  + Dataset to Dataset
* Actions (count, reduce, save, for each, collect, ...)
  + Eager evaluation
  + Dataset to Non-Spark format
* Eager operators are executed immediately