MapReduce

JOINS

Join

- Joins are relational constructs which are used to combine relations together.

Why Joins?

In MapReduce, joins are applicable in situations where you have two or more datasets you want to combine.

Eg:-

- An example would be when you want to combine your users with your log files that contain user activity details.
- ClickStream data might consist of Product Browse data, but the product details might be stored in another data on its own.
- Sales data might be a separate dataset, but product details might be stored elsewhere.

All of these above scenarios require you to join datasets together

Techniques of Joins in MR

There are mainly 3 broad kinds of joins in Mapreduce.

- Replication join—A map-side join that works in situations where one of the datasets is small enough to cache
- Semi-join—Another map-side join where one dataset is initially too large to fit into memory, but after some filtering can be reduced down to a size that can fit in memory
- Repartition join—A reduce-side join for situations where you’re joining two or more large datasets together

Map Side Join (Replicated Join)

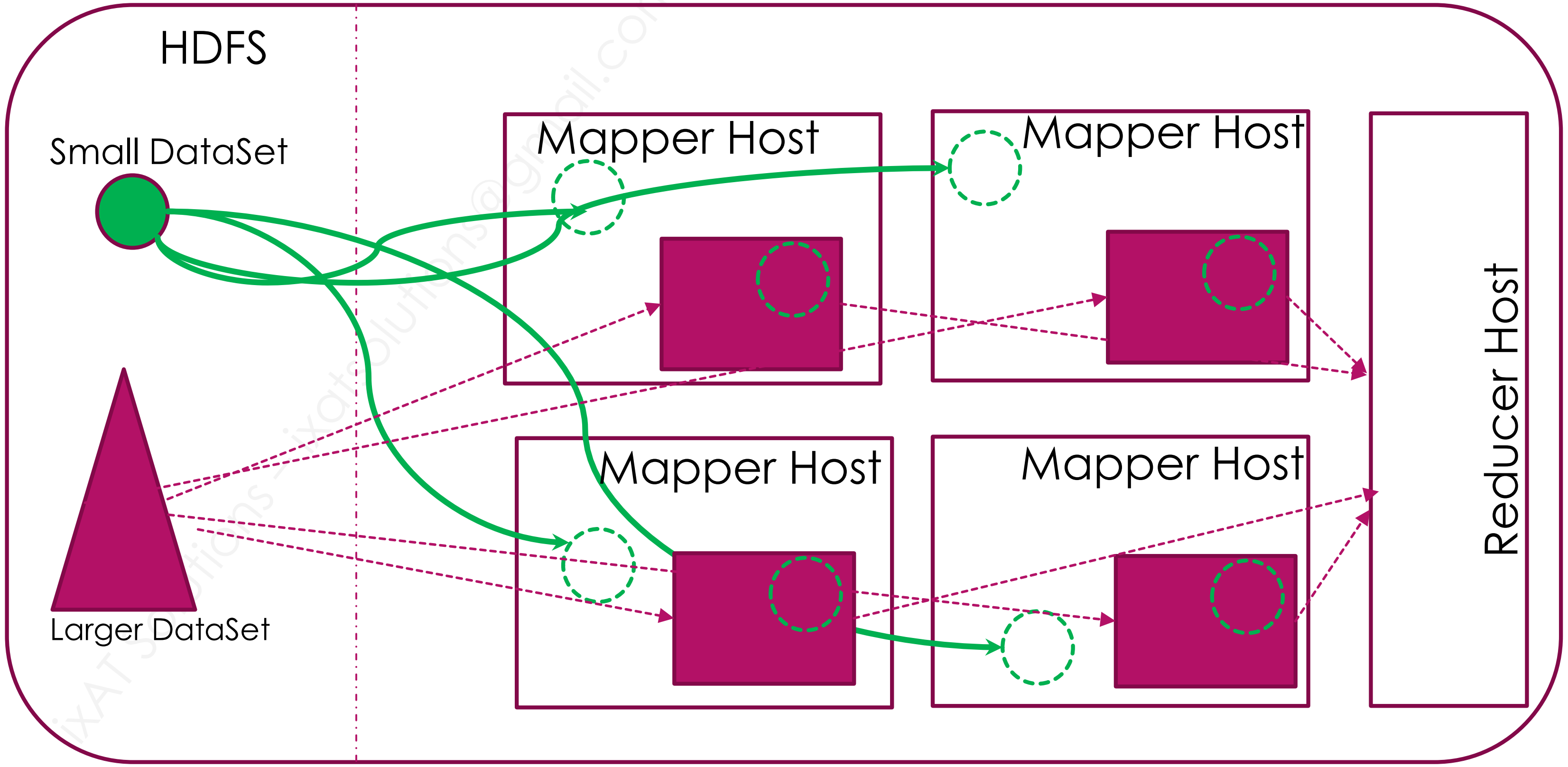
- A replicated join is a map-side join, and gets its name from its function, the smallest of the datasets is replicated to all the map hosts.
- The replicated join is predicated on the fact that one of the datasets being joined is small enough to be cached in memory.
- You'll use the distributed cache to copy the small dataset to the nodes running the map tasks, and use the initialization method of each map task to load the small dataset into a hashtable.
- Use the key from each record fed to the map function from the large dataset to look up the small dataset hashtable, and perform a join between the large dataset record and all of the records from the small dataset that match the join value.

MapSide Join – a feel

Driver

Driver would let Job know what DataSet need to be copied to all the Mapper Hosts. (this is called DistCache)

Mappers can read the small data set as if it is a local file, and prepare the data set in memory and use this as a lookup dataset to join with larger set.



Reduce Side Join

Joins of datasets done if done in the reduce phase are called reduce side joins.

- The key of the map output, of datasets being joined, has to be the join key - so they reach the same reducer
- Each dataset has to be tagged with its identity, in the mapper- to help differentiate between the datasets in the reducer, so they can be processed accordingly.
  - Tagging can be done by placing an Identifyer on Value
  - Also, one could do tagging if you use a Custom Writable with GenericWriter (fairly advanced).
- In each reducer, the data values from both datasets, for keys assigned to the reducer, are available, to be processed as required.
- If the input files are of different formats, we would need separate mappers, and we would need to use MultipleInputs class in the driver to add the inputs and associate the specific mapper to the same.

  [MultipleInputs.addInputPath(job, (input path n), (inputformat class), (mapper class n));]

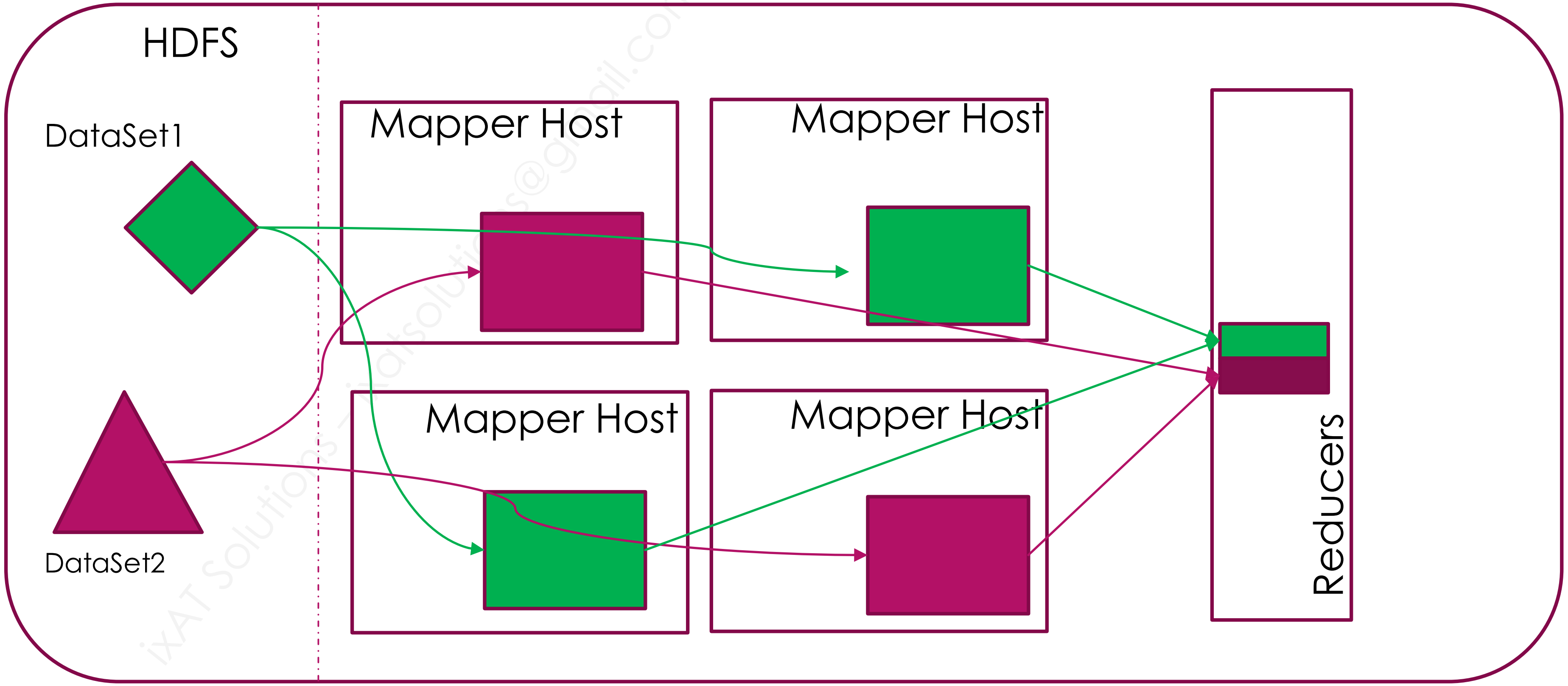
ReduceSide Join – a feel

Driver

Driver would prepare multiple mappers with different sets.

K,V's from mappers is multiplex and land into reducers.

Reducers need to differentiate between the value aggregates and do a join with the respective keys

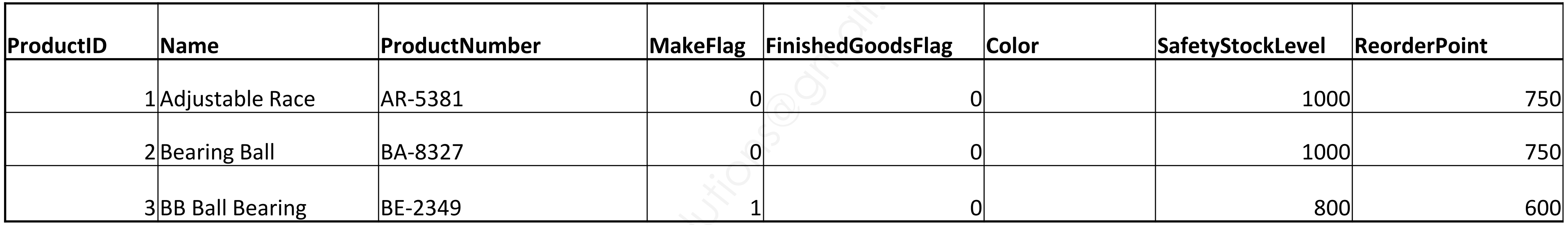


Sample Data

- http://msftdbprodsamples.codeplex.com/releases/view/55330
- Pick sample data from AdventureWorks 2012 OLTP Script
  - SalesOrderDetails.csv (5th column and 9th colum)
  - Product.csv (1st column and 2nd column)
  - Join key (Product ID , 9th column in Sales, 1st column in Product)

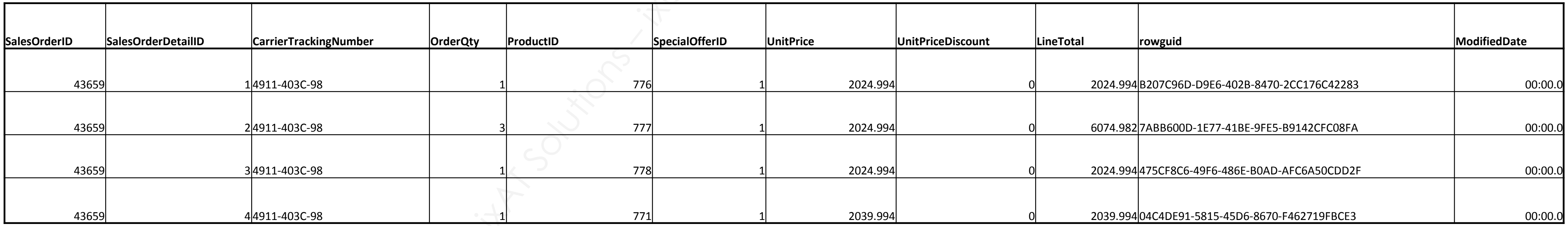
Schema

- Products Schema



ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint
1	Adjustable Race	AR-5381	0	0		1000	750
2	Bearing Ball	BA-8327	0	0		1000	750
3	BB Ball Bearing	BE-2349	1	0		800	600

Sales Schema



SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid	ModifiedDate
43659	1	4911-403C-98	1	776	1	2024.994	0	2024.994	B207C96D-D9E6-402B-8470-2CC176C42283	00:00.0
43659	2	4911-403C-98	3	777	1	2024.994	0	6074.982	7ABB600D-1E77-41BE-9FE5-B9142CFC08FA	00:00.0
43659	3	4911-403C-98	1	778	1	2024.994	0	2024.994	475CF8C6-49F6-486E-B0AD-AFC6A50CDD2F	00:00.0
43659	4	4911-403C-98	1	771	1	2039.994	0	2039.994	04C4DE91-5815-45D6-8670-F462719FBCE3	00:00.0