Good questions and explanation

<http://data-flair.training/forums/topic/list-the-advantage-of-parquet-file-in-apache-spark>

<http://data-flair.training/forums/topic/explain-transformation-and-action-in-rdd-in-apache-spark>

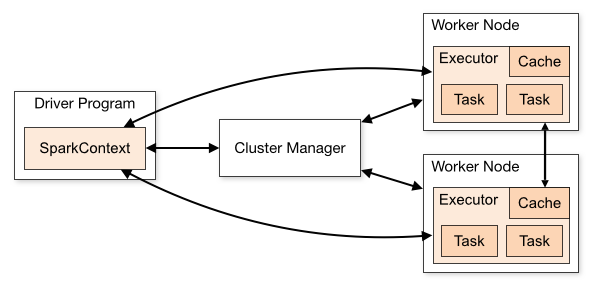
Spark Driver Program:

--------------------------------------

Driver program The process running the main() function of the application and creating the SparkContext.

Roll of Driver in Spark

* Driver program is responsible for launching various parallel operations on the cluster.
* **Driver program contains application's *main()* function.**
* **It is the process which is running the user code which in turn create the SparkContext object,**[**create RDDs**](http://data-flair.training/blogs/how-to-create-rdds-in-apache-spark/)**and performs**[**transformation and action operation on RDD**](http://data-flair.training/blogs/rdd-transformations-actions-apis-apache-spark/)**.**
* Driver program access [**Apache Spark**](http://data-flair.training/blogs/apache-spark-introduction-spark-comprehensive-tutorial/)through a [**SparkContext**](http://data-flair.training/blogs/sparkcontext-in-apache-spark-tutorial/) object which represents a connection to computing cluster (From Spark 2.0 onwards we can access SparkContext object through SparkSession).
* **Driver program is responsible for converting user program into the unit of physical execution called task.**
* It also defines distributed datasets on the cluster and we can apply different operations on Dataset (transformation and action).
* **Spark program creates a logical plan called**[**Directed Acyclic graph**](http://data-flair.training/blogs/directed-acyclic-graph-dag-in-apache-spark/)**which is converted to physical execution plan by the driver when driver program runs.**



What are the benefits of using parquet file-format in Apache Spark?

Parquet is a columnar format supported by many data processing systems. The benefits of having a columnar storage are -

1- Columnar storage limits IO operations.

2- Columnar storage can fetch specific columns that you need to access.

3-Columnar storage consumes less space.

4- Columnar storage gives better-summarized data and follows type-specific encoding.

Parquet is an open source file format for [**Hadoop**](http://data-flair.training/blogs/hadoop-introduction-tutorial-quick-guide/). Parquet stores nested data structures in a flat columnar format compared to a traditional approach where data is stored in row-oriented approach, parquet is more efficient in terms of storage and performance.

There are several advantages to columnar formats:

1)Organizing by column allows for better compression, as data is more homogeneous. The space savings are very noticeable at the scale of a Hadoop cluster.  
2)I/O will be reduced as we can efficiently scan only a subset of the columns while reading the data. Better compression also reduces the bandwidth required to read the input.  
3)As we store data of the same type in each column, we can use encoding better suited to the modern processors’ pipeline by making instruction branching more predictable.

**YARN**

---------------------

- Yet Another Resourec Negotiator (MapReduce-2)

- in MapReduce 1, Scalability a bottle neck when cluster size grows to 400+

- 2010 yahoo began the next generation MapReduce

- which can run different distributed processing frame work in parallel on the same cluster

-Main idea is to split the JOBTRACKER responsibilitys:

- Resource Manager - (Job Scheduling)

- Application Master - (Task Monitoring)

- older program written in MapReduce 1 work well with MapReuce2

- with MR2 only the way of execution of MR program changed

-so the program written in older api still works on MR2

Advantages-

---------------

- Increased Scalability: as the JobTracker task was split in to 2 scalibity increased dramatically

- More than one Yarn could co-exist on the same cluster.

along with MR we can have another distributed data processing framework (spark) on the same cluster

- better memory utilization with the concepts of containers

it is same as slots in classic MR - which are fixed in nature; where as containers are more flexible

in MR1 for single Task tracker would have fixed slots for map task and reduce task;

where as in containers it can run map/reduce or any other task and flexible in nature this results in better in memory utilization

Entities in yarn

---------------------

1. Client - responsible for submitting the job and interact with map and reduce and HDFS framework

2. Yarn Resource Manager - which is responsible for allocating the computing resources that are required by the job

job responsibilities can be classified in to 2

1. Scheduler - which responsible for scheduling of job; which does not perform monitoring/ tracking of job

2. Application Manager - which monitors the application status

3. Yarn Node Manager - it is present on all the slave node; responsible for launch and managing the containers

4. MR Application Master - it is responsible for execution of the job that is associated with;

- it is the one which coorbinates the task running and monitors the progress and aggregates it and sends the report to its client

- it is sponed(launched) under the Node Manager under the instructions of Resource Manager;

- it is sponed for ever job and terminates after the job completion

5.Yarn Child - it manages the execution of map and reduce task; responsible to send updates and progress to the application master

6. Distributed File System - which contains all necessary input and the place where the output files are returned to

Difference b/w MR1 and MR2

-------------------------------------------------

MR1 – default block size 64 MB

MR2 – Default block size is 128 MB

MR1

As a result Name node has to store less more info (meta data) in memory(RAM)

MR2

As a result Name Node has to store less meta data in memory(RAM) –