**What is the output of the program?**

- Estimate of the value of π using the Monte Carlo method based on the ratio of the number of points falling within a unit circle to the total number of generated points.

Output program: print("Pi is roughly %f" % (4.0 \* count / n))

**How does the program work?**

+ creates a Spark session.

+ generates random points.

+ calculates the **ratio of points** falling within a unit circle to the total number of points to estimate the value of π. (the **reduce function** with the add operator to calculate the sum of all the 1s and 0s, we obtain 2 lists of 1 and 0). (More dots in circle, more accuracy)

**What happens when the value of variable ‘slices’ increases from 1 to 2 and 4?**

If the **value of the variable slices increased** from 1 to 2 or 4**, number of partitions also increased** in which the data is distributed across the Spark cluster. In this case, the parallelize function is used to distribute the range of numbers across the specified number of slices.

When you increase the number of slices:

* slices = 2 -> two partitions -> computation is distributed across these two partitions.
* slices = 4 -> four partitions, computation is distributed across these four partitions.

Increasing the number of slices can **improve parallelism** and may lead to **faster computation, especially on a cluster with multiple nodes**. However, **consume more computation’s resources.**

**Explain the PySpark Environment**

The use of PySpark for distributed data processing and analysis. They provide the building blocks for expressing complex data workflows and optimizing the execution of tasks in a distributed environment.

1. **RDD (Resilient Distributed Dataset):**

RDD is the fundamental data structure in PySpark. It represents an immutable distributed collection of objects, partitioned across a cluster of machines.

+ Resilient: Recover lost data due to node failure by using lineage information.

+ Distributed: Data is distributed across multiple nodes in a cluster.

+ Immutable: Once created, RDDs cannot be changed, but new RDDs can be derived from existing ones through transformations.

Ex**:** rdd = sparkContext.parallelize([1, 2, 3, 4, 5])

1. **DataFrames (DF) and Datasets (DS):**

**DataFrame** is a distributed collection of data organized into named columns.It is an abstraction built on top of RDD, providing a higher-level, tabular structure. Supports various optimizations and is more user-friendly for SQL-like operations.

Ex**:** df = sparkSession.createDataFrame([(1, 'John'), (2, 'Jane')], ['ID', 'Name'])

**Dataset** is an extension of DataFrames with type-safety (available in languages like Scala and Java). Combines the benefits of DataFrames with the advantages of strongly-typed programming.

Ex**:** ds = df.as[(Int, String)]

1. **Transformation:**

Transformations are operations on RDDs, DataFrames, or Datasets that create a new distributed dataset from an existing one. Transformations are lazy, meaning they are not executed immediately but build up a logical execution plan.

Ex: map, filter, groupBy.

1. **Action:**

Actions are operations that trigger the execution of transformations and return values to the driver program or write data to an external system. They kick off the computation and force the evaluation of the transformations.

Ex: count, collect, and saveAsTextFile.

1. **Lineage:**

Lineage is the sequence of transformations that were applied to create an RDD. It allows RDDs to recover lost data by recomputing only the affected partitions, making the system fault-tolerant. Helps in achieving fault tolerance and data recovery in case of node failures.

1. **Partition:**

A partition is a basic unit of parallelism in RDDs, representing a portion of the dataset. RDDs are divided into partitions that are processed concurrently across multiple nodes in a cluster. The number of partitions affects the parallelism of the computation.

Ex**:** When creating an RDD using parallelize or through certain transformations, you can specify the number of partitions.