World population growth rate project from 1970 to 2022

By

Mahin Ahmed
Id: 191-15-1029

&

Md.Jannatul Ferdous
Id: 191-15-2497

&

Saiful Islam Sagor
191-15-2678

&

Md.Omer Faruk Tusher
192-15-13122

Department of Computer Science & Engineering

Daffodil International University, Dhaka, Bangladesh.

Emails: mahin15-1029@diu.edu.bd, jannatul15-2497@diu.edu.bd, Saiful15-2678@diu.edu.bd, omer15-13122@diu.edu.bd

Abstract:

The purpose of our project is to predict the populations of different countries over the world. Global population growth is a challenging factor for the human race.

Distributed data processing platforms for cloud computing are important tools for large-scale data analytics. Apache Hadoop MapReduce has become the de facto standard in this space, though its programming interface is relatively low-level, requiring many implementation steps even for simple analysis tasks.

The main aim of this project is to analyze and predict the massive amount of data (world population), with the help of various types of tools such as apache spark which is used for real-time processing and analysis of large amounts of data.

Brief description of the project:

The project we're work on "World Population Database" which is a real-world dataset. This is project built up by using pySpark and Python. The methods used in regression, linear regression and logistic regression, etc.

From this dataset, we can figure out the total population of each country in the world from 1970 to 2022. Also, we'll know the rate of birth, increasing and decreasing, and predict the percentages of the population of each country all over the world from 1970 to 2022.

Big data problems have brought many changes in the way data is processed and managed over time. Today, data is not just posing a challenge in terms of volume but also in terms of its high-speed generation. The data quality and validity varies from source to source, and thus are difficult to process.

Experiments:

In our project, we will predict and analyze the total growth rate of the world population. So we need to do a few experiments by using big data analysis processes. There have various tools for controlling the massive data in big data such as apache spark (which we will use in our project). Because apache Spark is a popular open-source distributed data processing framework that can efficiently process massive amounts of data. It provides more than 180 configuration parameters for users to manually select the appropriate parameter values according to their own experience.

Spark proposed two important terms: Resilient Distributed Datasets (RDD) and Directed Acyclic Graph (DAG). These two techniques work together perfectly and accelerate Spark up to tens of times faster than Hadoop under certain circumstances, even though it usually only achieves a performance two to three times more quickly than MapReduce. It supports multiple sources that have a fault tolerance mechanism that can be cached and supports parallel operations. Besides, it can represent a single dataset with multiple partitions.

When Spark runs on the Hadoop cluster, RDDs will be created on the HDFS in many formats supported by Hadoop, likewise text and sequence files. The DAG scheduler system expresses the dependencies of RDDs. Each spark job will create a DAG and the scheduler will drive the graph into the different stages of tasks then the tasks will be launched to the cluster. The DAG will be created in both maps and reduced stages to express the dependencies fully. When users submit their applications to a Spark cluster, they expect to process a certain amount of data within an execution time target. Some applications may have tight deadlines, while others may have more flexible deadlines.

As described previously, Spark's default resource allocation method executes all submitted applications in a FIFO order, and each application uses all available cluster resources This approach prevents resource sharing and can negatively affect applications with tight deadlines. Therefore, Spark users and job schedulers prefer to use a different resource allocation mechanism, such as YARN that allows resource sharing. In this mode, Spark users or job schedulers can specify how many executors, cores, and memory an application can have. A fast and accurate execution time prediction model is needed to ensure that users and

schedulers request just the right amount of resources so as to meet deadlines in a cost-effective manner.

Related works:

Experimenting with Hadoop Fault-Tolerance, we'll also work on Hadoop frameworks to test real-world faults' implications on MapReduce applications in our project dataset. Fault detection is the first building block of a fault-tolerant framework that desires to detect faults as soon as they occur. Hadoop MapReduce uses heartbeat monitoring based on the push model as a default approach.

Conclusion:

Mainly our project is based on a real-time dataset called "World Population". By using this dataset we can predict and analyze the total growth rate (from 1970 to 2022) of each country in the world and rank it in ascending order. While we calculate the total growth rate also have to predict and analyze the area(kmÂ²) and density(kmÂ²).
In our dataset, we'll work with apache-spark, also using regression for prediction the growth rate of 2023.