Shyam Patel
November 24, 2019

CS441 HW3: Monte Carlo Simulation

Background

This project aims to predict stock portfolio losses using Monte Carlo simulation and, in order to do so, utilizes the open-source Apache Spark 2.4.4 cluster-computing framework to program clusters with implicit data parallelism on the Amazon EMR cloud data platform. This Spark application takes advantage of resilient distributed datasets (RDDs), immutable read-only, fault-tolerant distributed collections of objects that are divided into logical partitions and may be computed on different nodes of the cluster. This allows for efficient parallel processing of the data, which would otherwise take much longer to process. The RDDs are created in two ways—either by *referencing* a dataset stored in the Hadoop Distributed File System (HDFS), as is the case when the application uses the `change.csv` and `portfolio.csv` datasets, or by *parallelizing* an existing collection in the driver program, which is what occurs when calling the `runSession` method during each run. When the results RDD is obtained, the mean, standard deviation and 5th, 25th, 50th, 75th and 95th percentiles are computed for the funds values. When the application is run locally, the master is set to run with as many worker threads as logical cores on the machine using the `local[*]` tag.

Monte Carlo

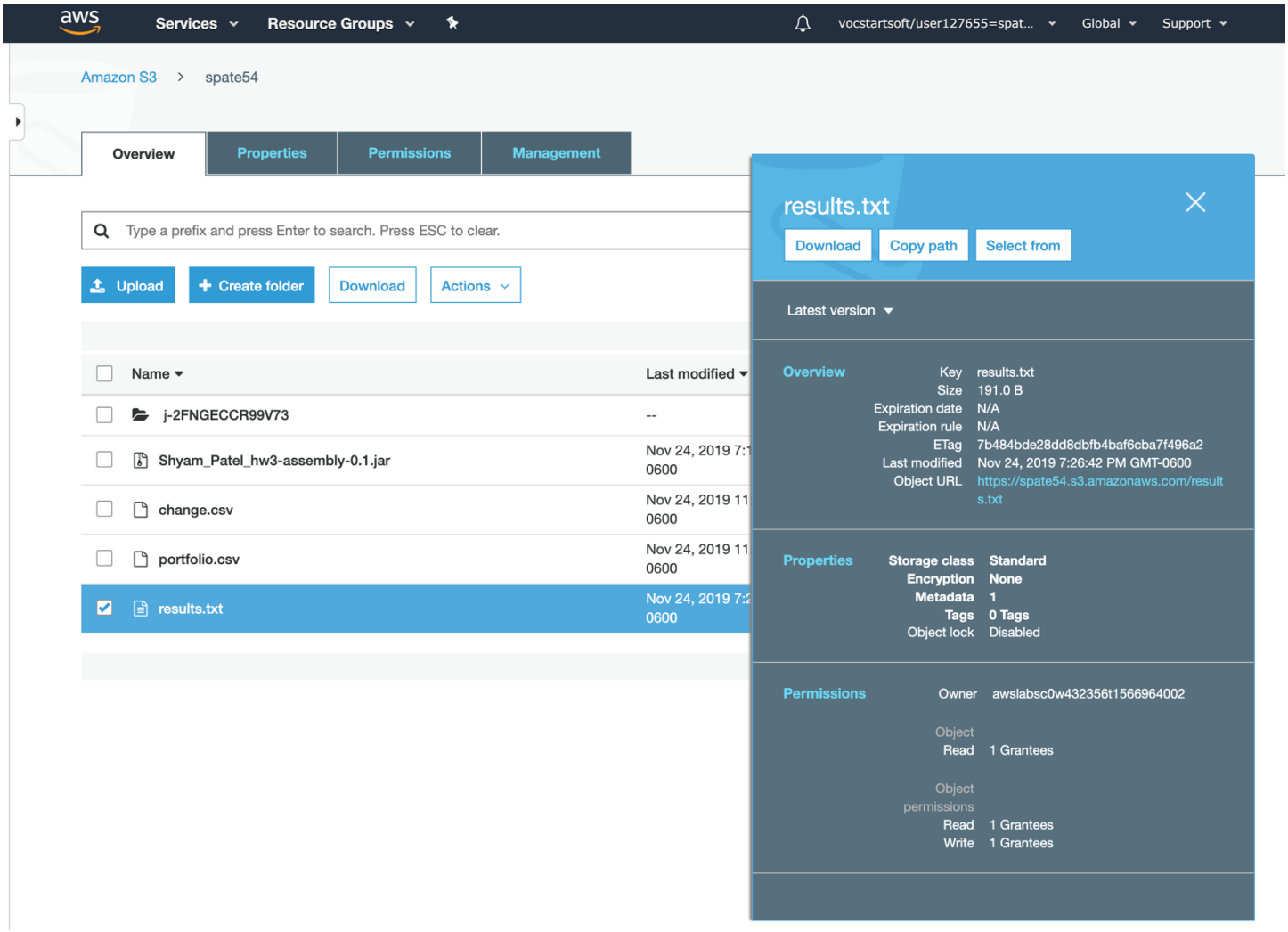
The Monte Carlo simulation in this project consists of 10,000 runs and 500 partitions. This is accomplished using the RDD parallelize function. In each run, the following tasks are implemented.

1. The historical stocks data is in the form of percent change of the close price for each day of the companies Google (GOOGL), Amazon (AMZN), Facebook (FB), Apple (AAPL) and Microsoft (MSFT), ranging from May 2012 to the present. The data is compiled using the `download.py` Python script that uses the Alpha Vantage API located in the `src/main/data directory`, the result of which is stored in `src/main/data/change.csv`.
2. The initial portfolio used in each simulation is consistent and consists of ticker symbols and the funds allocated for each in USD. This is located in `src/main/resources/portfolio.csv`.
3. In each simulation run in parallel, gains and losses are computed daily for each investment that is part of the portfolio. The subset of values is shuffled to mimic market forces. These gains and losses are recorded in the portfolio.
4. For each day in which there is a net gain in the portfolio, a decision is made to buy another stock. The stock is selected in random based on its performance for that day. Only stocks that had a net positive change in close price for the day are considered for investment.
5. For each day in which there is a net loss in the portfolio, a decision is made to possibly sell the stock that had the most negative performance that day. In this case, the investment of the stock that is sold is transferred to the stock that performed most exceptionally that day. If no such stock exists, or it is the same stock that is being considered for sale, then the buying and selling is simply not performed.
6. This process repeats itself until all the values in the shuffled subset are exhausted.

Shyam Patel
November 24, 2019

Results

The results of the Spark application can be obtained using the Amazon S3 portal. From the S3 bucket directory, select `results.txt` and click the Download button.



The following are a sample of results obtained from running the simulation on Amazon EMR.

```
Mean:              $7113.54
Std deviation:     5335.372
5th  percentile:   $1572.98
25th percentile:   $2663.25
50th percentile:   $5292.40
75th percentile:   $10518.66
95th percentile:   $18186.97
```

As can be observed, the mean investment at the end of the Monte Carlo simulation (spanning ~7 years) have increased sharply from the initial $1,250 investment recorded in portfolio.csv. The funds vary based on the buying and selling prescribed by the algorithm that takes place through each run of the simulation. The overall takeaway is that the funds tend to gain over time.