**ANALYSIS REPORT**

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

|  |  |
| --- | --- |
| **Version:** | **1.1** |
| **Status:** | Draft |
| **Approver:** | Ngô Thái Bình  Nguyễn Thị Diễm Trang |
| **Author:** | Quách Hoàng Minh  Nguyễn Vũ Anh Thư Nguyễn Bảo Nguyên |
|  |  |



**Document history**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Status | Remarks |
| 1.0 | 24-02-2022 | Nguyễn Bảo Nguyên | Draft |  |
| 2.0 | 02-05-2022 | Nguyễn Vũ Anh Thư Quách Hoàng Minh | Edit 1 |  |

**References**

|  |  |  |  |
| --- | --- | --- | --- |
| Reference | Title | Author | Version |
|  |  |  |  |

Table of contents

[1. Problems 4](#_Toc96716144)

[2. Experiments and Results 4](#_Toc96716145)

[2.1. One stock at a time 4](#_Toc96716146)

[2.2. Sequential two stocks 4](#_Toc96716147)

[2.3. Sequential two stocks with interval and release resources 4](#_Toc96716148)

[2.3.1 Interval = 5 4](#_Toc96716149)

[2.3.2 Interval = 15 4](#_Toc96716150)

[3. Conclusion 5](#_Toc96716151)

# Problems

When executing the script to predict the price of 300 shares sequentially on the server, it stuck the CPU, leading to the server crash.

# Experiments and Results

## Delay time when prediction

### One stock at a time

* Peak memory usage is 386.58MB
* Peak CPU utilization is 92.9 %
* Execution time is 29.82s



Figure : Memory Usage when run one stock at a time

### Sequential two stocks

* Peak memory usage is 460.82MB
* Peak CPU utilization is 100.0 %
* Execution time is 51.0s



Figure : Memory usage when run sequentially 2 stocks

### Sequential two stocks with interval and release resources

### Interval = 5

* Peak memory usage is 490.76MB
* Peak CPU utilization is 93.9 %
* Execution time is 61.09s



Figure : Memory usage when run sequentially 2 stocks with interval =5 and release resources

### Interval = 15

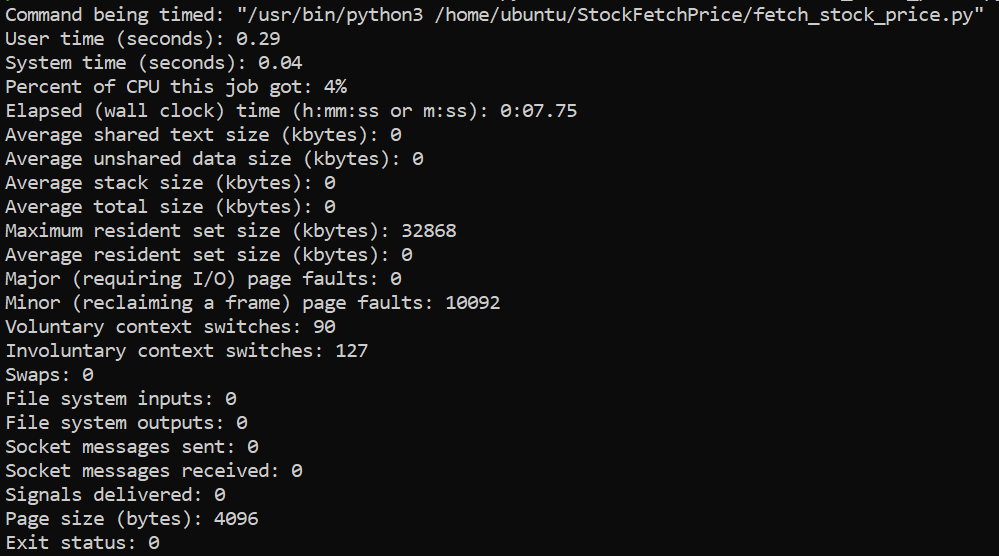
* Peak memory usage is 490.92MB
* Peak CPU utilization is 98.9 %
* Execution time is 81.46s



Figure : Memory usage when run sequentially 2 stocks with interval = 15 and release resources

## Delay time when fetch data

### Fetch stock price time and memory usage

* Peak memory usage is 32868kB
* Peak CPU utilization is 4 %
* Execution time is 0.29s

### Query for prediction time and memory usage

* Peak memory usage is 70356kB
* Peak CPU utilization is 51 %
* Execution time is 0.48s

### 

### Fetch company information time and memory usage

## 

# Conclusion

Based on the results of the previous attempts, we decided to run stocks sequentially, releasing resources after completing one stock and sleeping for 5 seconds after running the next. We tried to run 300 stocks continuously, however as we reached roughly 83 stocks, the server crashed. To avoid CPU overloading, we divided the 300-stock list into six smaller lists, each with 50 stocks. And since each stock costs between 35 and 50 seconds (including sleeping 5s), a total of 50 stocks will cost around 2500 seconds. We depend on it to set up a cronjob to execute the script automatically, with each list (50 stocks) running 1 hour 30 minutes apart to avoid CPU overloading.