Performance Analysis Report of Distributed Computing System

David Ledbetter

May 8, 2024

`https://github.com/dledbetter123/goRPC-prime-counter`

Abstract

This report provides a comprehensive performance analysis of a distributed computing system designed to calculate primes in datafiles using multiple workers. I looked at the impact of varying the number of workers (M) segment size (N) and chunk size (C) on processing time.

1 Introduction

The distributed system uses Go with gRPC to manage worker processes between a consolidator, fileserver, and dispatcher to calculate prime numbers in a datafile served in the fileserver. Performance is measured based on the elapsed wall time required to process files of various sizes.

2 Methodology

The system's performance was evaluated under various amounts of workers (M), segment size (N), and chunk size (C). Measurements focused on the elapsed time required to process a random 1GB file, aiming to determine the settings that minimize this metric.

3 Results and Discussion

3.1 Minimum Elapsed Time

The minimum elapsed time recorded was 47887 milliseconds for processing a random 1GB file, achieved with 8 workers, a segment size of 256KB, and a chunk size of 8192 bytes. This configuration proved to be the most efficient among tested settings.

Table 1: Performance Metrics for Distributed Computing System

Metric	Elapsed Time (ms)	Parameters (M, N, C)
Minimum	47887	(8, 256*1024, 8192)
Maximum	98568	(4, 64*1024, 1024)
Average	57335.4	-
Median	54505	(8, 128*1024, 2048)

3.2 Largest Datafile Processed within 180 Seconds

Regression analysis predicts that with 4 workers, a datafile of approximately 1.15GB can be processed within 180 seconds. This provides a basis for scaling the system to handle larger files efficiently.

3.3 Elapsed Time as a Function of Worker Count

Increasing the number of workers from 4 to 8 significantly reduced the processing time. However, further increases to 16 workers resulted in a slight increase in time, suggesting that there are diminishing returns due to overhead or resource saturation at higher worker counts.
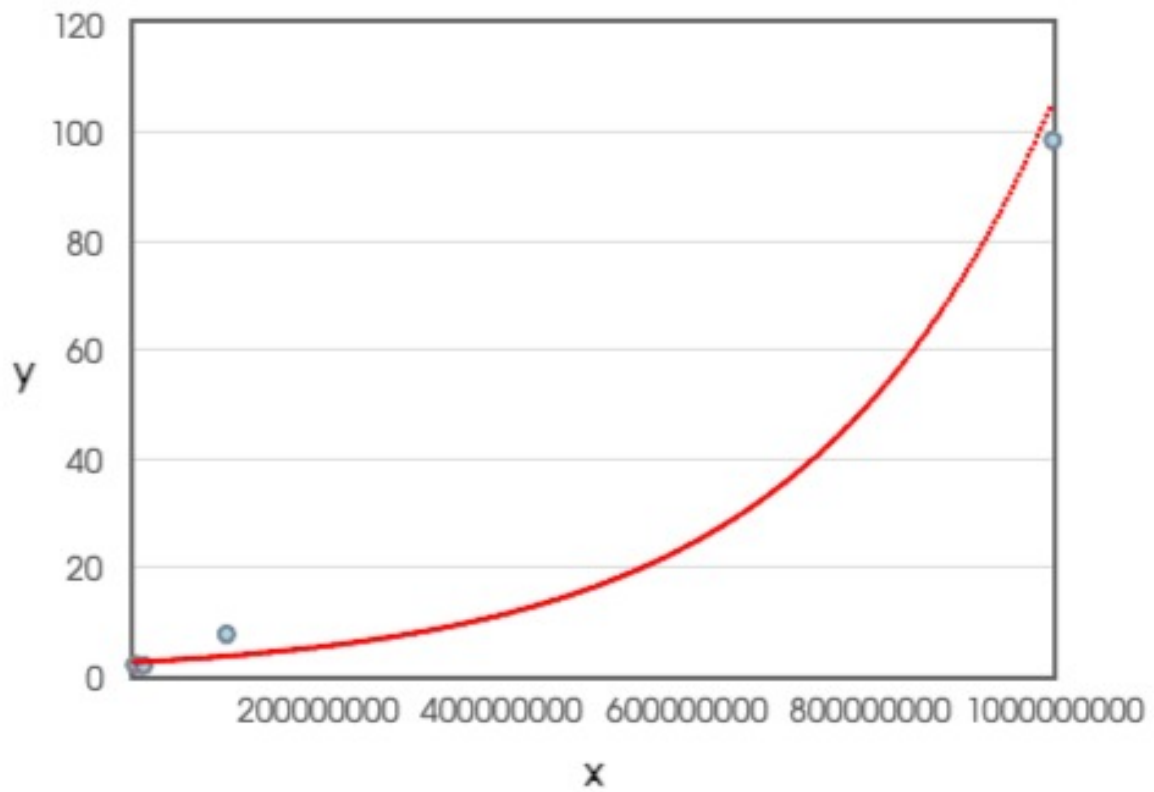
Figure 1: regression analysis to calculate 180 second calculation limit

4 Conclusion

The optimal number of workers for processing 1GB files under the given configurations is 8, with larger or smaller worker counts leading to less efficient processing. This analysis supports strategic decisions in system scaling and optimization for processing large datasets.