A Comparison of String Search Algorithms for Deep Packet Inspection

Kieran Hunt

September 2, 2015

Abstract

Contemporary deep packet inspection (DPI) systems often rely on custom hardware or entrenched ideas about the string search algorithms used. These algorithms have mathematically provable time or space complexities however not much is empirically known about their performance on real-world packet datasets. We felt that some string search algorithms could produce results that differed from their theoretical performance within the context of packet inspection. Furthermore, we sought to show that even algorithms with similar theoretical performances could produce differing practical results. Our approach was to reimplement a variety of the established string search algorithms and run them through a diverse set of tests with both real-world and constructed datasets. Our tests found that the Bloom filter was the fastest overall, Rabin-Karp was the most memory efficient and that the Nave algorithm was the slowest. Furthermore we found that, although the Bloom and Cuckoo filters have the same theoretical time complexity, the cuckoo filter was almost twice as slow as its counterpart. These findings help to show which algorithms perform best in practice and can help future algorithm designers to improve on the current approaches. In practice we show which algorithms a designer of a deep packet inspection system should consider implementing based on our findings.

1 Introduction

Here is the text of your introduction. [1]

1.1 Subsection Heading Here

Write your subsection text here.

2 Algorithms

Write your conclusion here.

3 Testing Environment

Write your conclusion here.

4 Results

Write your conclusion here.

5 Analysis

Write your conclusion here.

6 Conclusion

Write your conclusion here.

References

[1] R. Karp and M. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development 31.2*, 1987.