FCN: URL Shortening Services (USS)

Akshay Nehe and Vamshi Muthineni

Topics

Introduction

Project Objectives

USS - Components and Design

Demo

Experiments

Member Contributions

Introduction

- URL Shorteners are used extensively: social media, SMS, advertisements, etc;
- Pros: USS improve readability and convenience;
- Cons: Redirection obscures target and can trick users;

Project objective

- Study components of a URL Shortener;
- Explore the techniques to make USS more secure and quantify the secureness;
- Analyze the usage of USS by users;

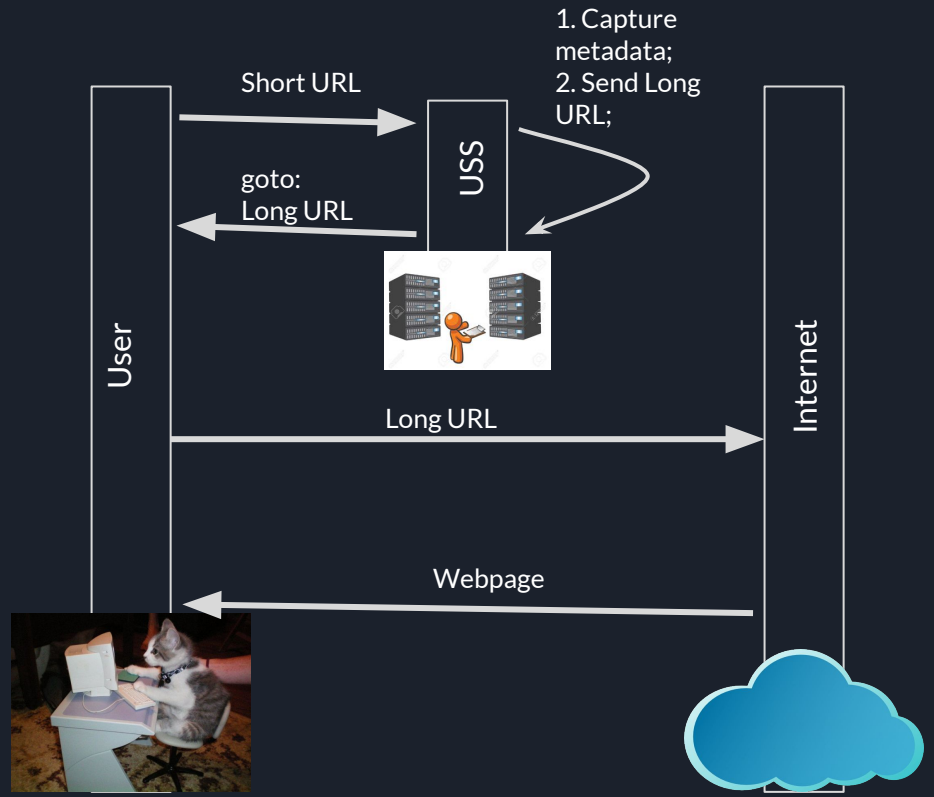
USS Design

- Bijective function:

  F(long_URL) ←→ short_url

- Short URL must be unique

- Attributes:
  - URL length, Ex 3 or 6;
  - Character set, Ex: [a-zA-Z0-9];



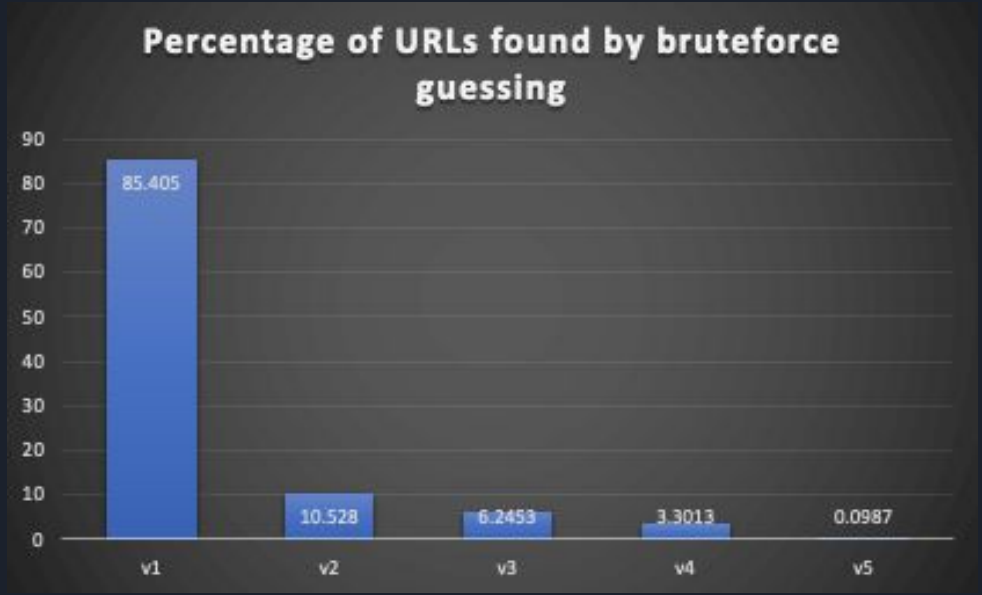
Demo

URL: 172.24.20.134:8080

Github Repo: https://github.com/invincible-akshay/url-shortener

Experiments I

Ver	Description	Hits
1	len = 3 charset = a-z	12833
2	len = 3 charset = a-zA-Z	1598
3	len = 3 charset = a-zA-Z0-9	950
4	len = 4 charset = a-z	521
5	len = 4 charset = a-zA-Z0-9	22

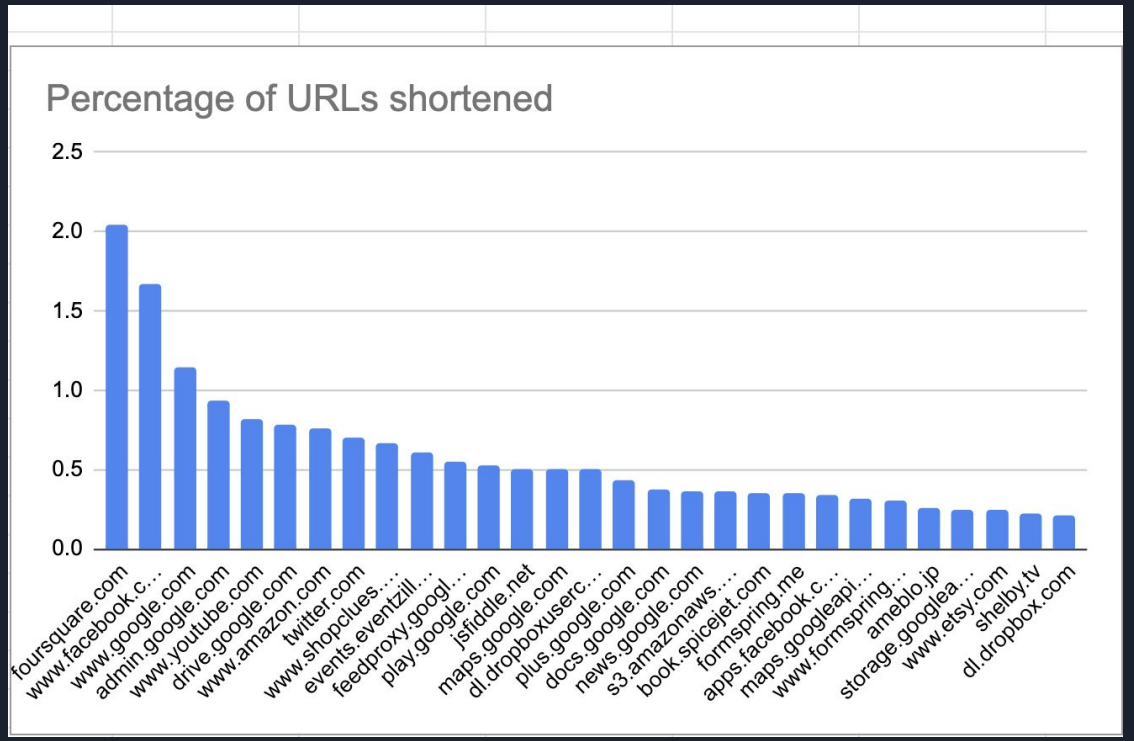


Ex:

1. url_size 4, character_set_size 26 = 0.45 M,
2. url_size 4, character_set_size 62 = 7.3 M,
3. url_size 5, character_set_size 26 = 11 M

Experiments II

- Shortened URL Analysis
- Data dump -> preprocessing
- -> mapreduce cluster -> analysis/results
- Top url domains which were shortened
- Further experiments - redirection code



Conclusions

- The shorter the token, easier to brute force. And if the token is too large, loses the convenience of short url.
- Trade-off between convenience and security
- One other way to keep short urls secure is by expiring urls.
- Timely cleaning up to avoid link rot.
- Generated tokens/urls should be uniformly distributed over the range. Ex: bit.ly

Member Contributions

- Literature Survey - both
- URL Shortener implementation - Akshay
- Brute forcing the URL shortener - Vamshi
- Metrics generation on existing shortened URLs - Vamshi

Thank you!

Questions?