

URL shortening service

Goal: service provides short aliases redirecting to long URLs

Step 1: Requirements

Functional:

- Given URL, service should generate shorter and unique alias of it
- Service should redirect short link to original
- User can choose custom link
- Links expire after a standard timespan

Non-functional:

- Highly available
- Minimal latency for redirection

Extended:

- Analytics e.g., how many redirections
- Accessible via REST APIs by other services

Step 2: Back of Envelope

Traffic estimates:

- Read-heavy system assume 100:1 read to write ratio
- 500M shortenings per month results in $100 \times 500M = 50B$ redirections per month
- 500M per month \rightarrow 200 new URL/s
- 100×200 new URL/s \rightarrow 20K/s URL redirections/s

Storage estimates:

- Store every URL for 5 years $\rightarrow 500M \times 5y \times 12m = 30B$
- Each object is 500b $\rightarrow 30B \times 500b = 15TB$

Bandwidth estimates:

- Writes: $200 \text{ URL/s} \times 500b = 100KB/s$
- Reads: $20K \times 500b = 10MB/s$

Memory estimates:

- Cache hot URLs assuming 80-20 rule i.e., 20% of URLs generate 80% traffic
- $20K/s \times 3600s \times 24h = 1.7B \text{ req/d} \times 0.2 \times 500b = 170GB$

Step 3: System API

- createURL(api_key, orig_url, cust_alias, uname, exp)
- deleteURL(api_key, url_key)