**Designing a URL Shortening service like TinyURL**

* Why do we need URL shortening?

Ans :- It is required to shorten the long URLs.

* Requirements and Goals of the Systems

Ans:-

1. **Functional Requirements**
2. Given URL, our service should generate a shorter and unique alias of it. This is called a short link.
3. When users access short link, it should redirect to original link.
4. Users should have flexibility of picking custom short link for their URL.
5. Link will expire after a standard default timespan.
6. **Non-Functional Requirements**
7. The system should be available.
8. URL redirection should happen in real-time with minimal latency.
9. Shortened links should not be guessable.

**Extended Requirements:**

1. Analytics e.g. how many times a redirection happened?
2. Our service should be accessible through REST APIs by other services.

**3. Capacity Estimation and Constraints**

Our System will be read-heavy. There will be a lot of redirection compared new URL shortenings. Let’s assume 100:1 ratio between read and write.

**Traffic estimates:** Assuming, we will have 500M new URL shortenings per month, with 100:1 read/write, we expect 50B redirections during the same period.

What would be Queries Per Second(QPS) for our System? New URLs shortenings per second:

500 Million/ (30 days\*24 Hours\*3600 Seconds) = ~200 URLs/s.

**Storage estimates:** Let’s assume we store every URL shortening request (and associated shortened request (and associated shortened link) for 5 years. Since we expect to have 500M new URLs every month, the total number of objects we expect to store will be 30 billion:

500 million \* 5 years \* 12 months = 30 billion

Let’s assume that each stored object will be approximately 500 bytes (just a ballpark). We will need 15TB of total storage.

30 billion \* 500 bytes = 15 TB

**Bandwidth estimates:**

For write requests, since we expect 200 new URLs every second, total incoming data for our service will be 100KB per second:

200\*500 bytes = 100KB/s

For read requests, since every second we expect ~20K URLs redirections, total outgoing data for our service would be 10MB per second:

20K \* 500 bytes = ~10 MB/s

**Memory Estimates:**

If we want to cache some of the hot URLs that are frequently accessed,