**Web Crawler:** If you were designing a web crawler, how would you avoid getting into infinite loops?

**Solution:**

Normally a crawler will take a list of most popular website and then crawl the page for its contents and then record the list of URL into the queue and then proceed to the next in queue. In this way the loop can occur if the queue is loaded with the URL already crawled by the bot.

Now the loop could be avoided by seeking the database if the current URL/Content is already been visited. If not then we can crawl and then record the findings and proceed with the next set

**Increasing Speed:**

The parallel crawling of robot could speed up the process and we need to handle the concurrent update of the database by using mutex/lock mechanism

On the database side the huge set of data instead of being kept in a single one can be distributed with the use of **sharding** with the hashing as it will create a fetching much easier. However the adding of new database and so re-indexing will be a hectic one that should be addressed

**Idea:**

Not sure if the following is possible but we can create a **Master** and the **Worker** threads that will do the web crawling and the master guides the set of targets for the workers which will ensure no duplicates are being crawled and no waiting time/ issue occur.

Drawbacks would be a single point of failure to the Master; routing other worker thread to do the work if the actual worker has become slow; asking multiple thread to do the same work and discard once any one thread completes