Functional Requirements

- Should be able to parse html content and extract urls from a web page.
- Should be able to do malware detection.
- Should be able to detect duplicate urls.
- repeat the whole process infinitely.

Non Functional Requirements

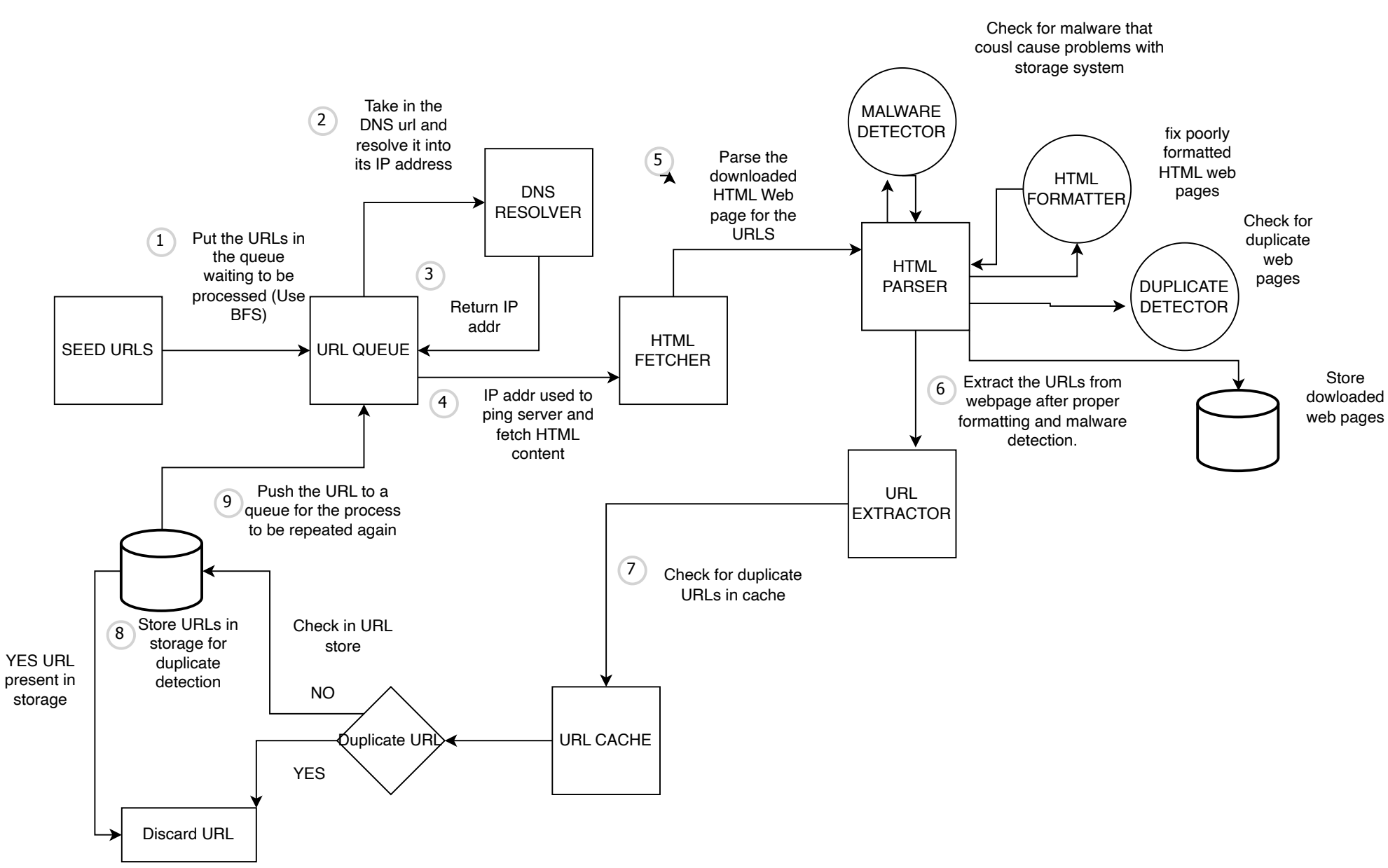
- Web crawler should be scalable because there are billions of pages to be scaled. Web crawler should efficiently use parallelization.
- Should not overwhelm the website servers with too much requests otherwise it would lead to DDOS attacks.
- Robust enough to handle poorly formatted HTMLs, unresponsive servers and web crashes.
- Ability to include future changes, flexible and extensible enough.
- Should have a robust logging and monitoring feature to keep a track of metrics and overall health of web crawler.

QPS Estimation

- Assuming 1 billion pages to be crawled every month.
- 1000000000 / 30 days / 24 hours / 60 minutes / 60 seconds = 385 QPS
- Peak value assumed to be a little more than twice that is 800 QPS.

Storage Estimation

- Assuming average page size is 500 KB,
- 1 billion pages require 500 TB storage a month
- If the pages need to be stored for a year , then we require 500 TB * 12 = 6 PB of storage a year.


Check for malware that cousl cause problems with storage system
MALWARE DETECTOR
2
Take in the DNS url and resolve it into its IP address
DNS RESOLVER
5
Parse the downloaded HTML Web page for the URLS
HTML FORMATTER
fix poorly formatted HTML web pages
1
Put the URLs in the queue waiting to be processed (Use BFS)
3
Return IP addr
HTML PARSER
Check for duplicate web pages
DUPLICATE DETECTOR
SEED URLS
URL QUEUE
HTML FETCHER
4
IP addr used to ping server and fetch HTML content
6
Extract the URLs from webpage after proper formatting and malware detection.
Store dowloaded web pages
URL EXTRACTOR
9
Push the URL to a queue for the process to be repeated again
7
Check for duplicate URLs in cache
8
Store URLs in storage for duplicate detection
Check in URL store
YES URL present in storage
NO
Duplicate URL
URL CACHE
YES
Discard URL