CS317

Information Retrieval

Week 09

Muhammad Rafi

April 7, 2021

Web Crawler

Chapter No. 20

Web Crawler

- Web crawling is the process by which we gather pages from the Web to index them and support a search engine.
- The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them.
- web crawler is sometimes referred to as a spider.

Feature a Crawler MUST provide

- Robustness: The crawler must be robust to deal with a large number of linked pages from a website. Sometime server traps a crawler, the crawler must identify these traps.
- Politeness: Web servers have both implicit and explicit policies regulating the rate at which a crawler can visit them. These politeness policies must be respected.

Feature a Crawler Should provide

- Distributed: The crawler should have the ability to execute in a distributed fashion across multiple machines.
- Scalable: The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth.
- Performance and efficiency: The crawl system should make efficient use of various system resources including processor, storage, and network bandwidth.

Feature a Crawler Should provide

- Quality: Given that a significant fraction of all web pages are of poor utility for serving user query needs, the crawler should be biased toward fetching “useful” pages first.
- Freshness: In many applications, the crawler should operate in continuous mode: It should obtain fresh copies of previously fetched pages.

Feature a Crawler Should provide

- Extensible: Crawlers should be designed to be extensible in many ways – to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture be modular.

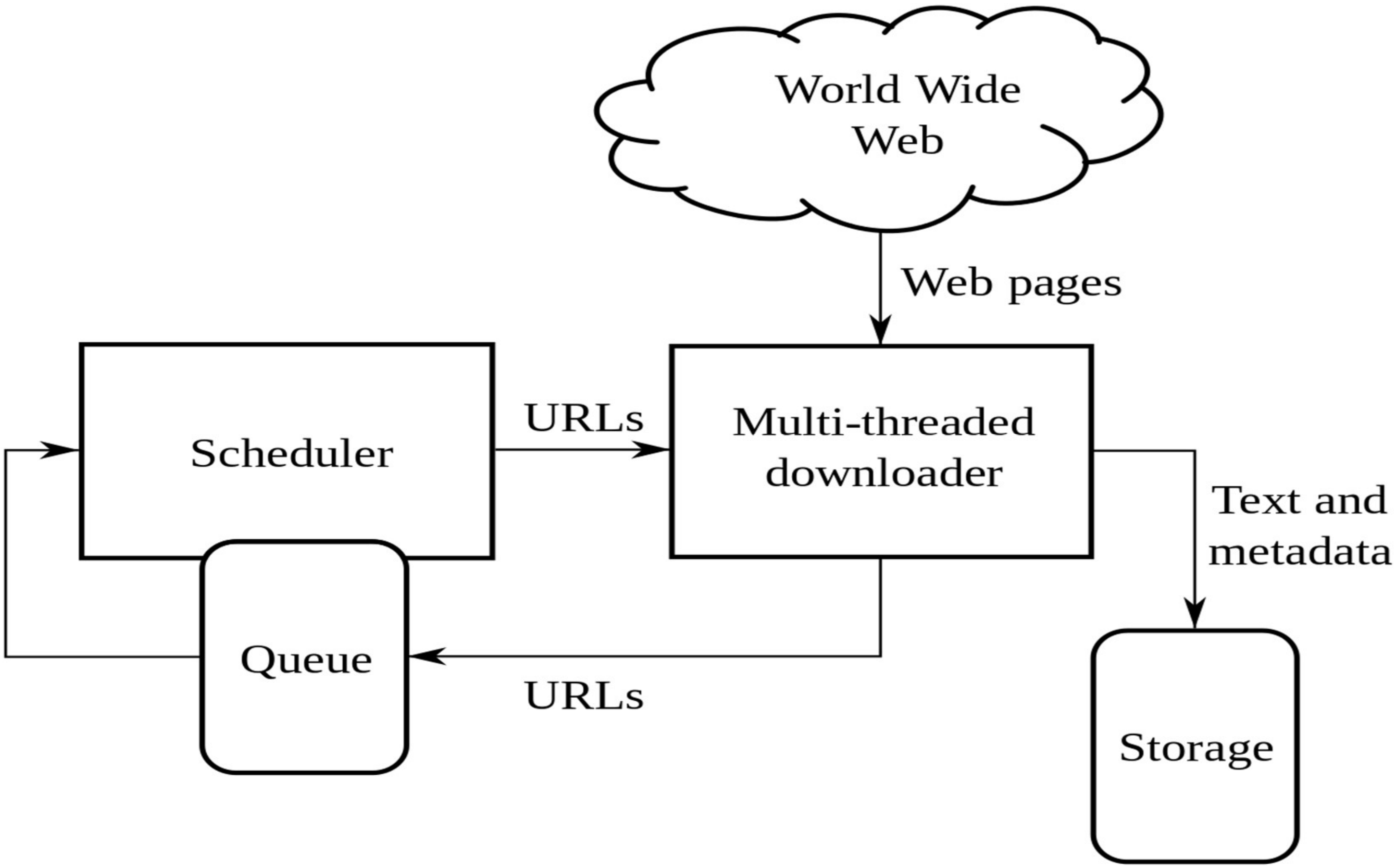
Different Type of Crawlers

- Path-ascending Crawler
- Topic Focused Crawler
- Academic Focused Crawler
- Semantic Focused Crawler

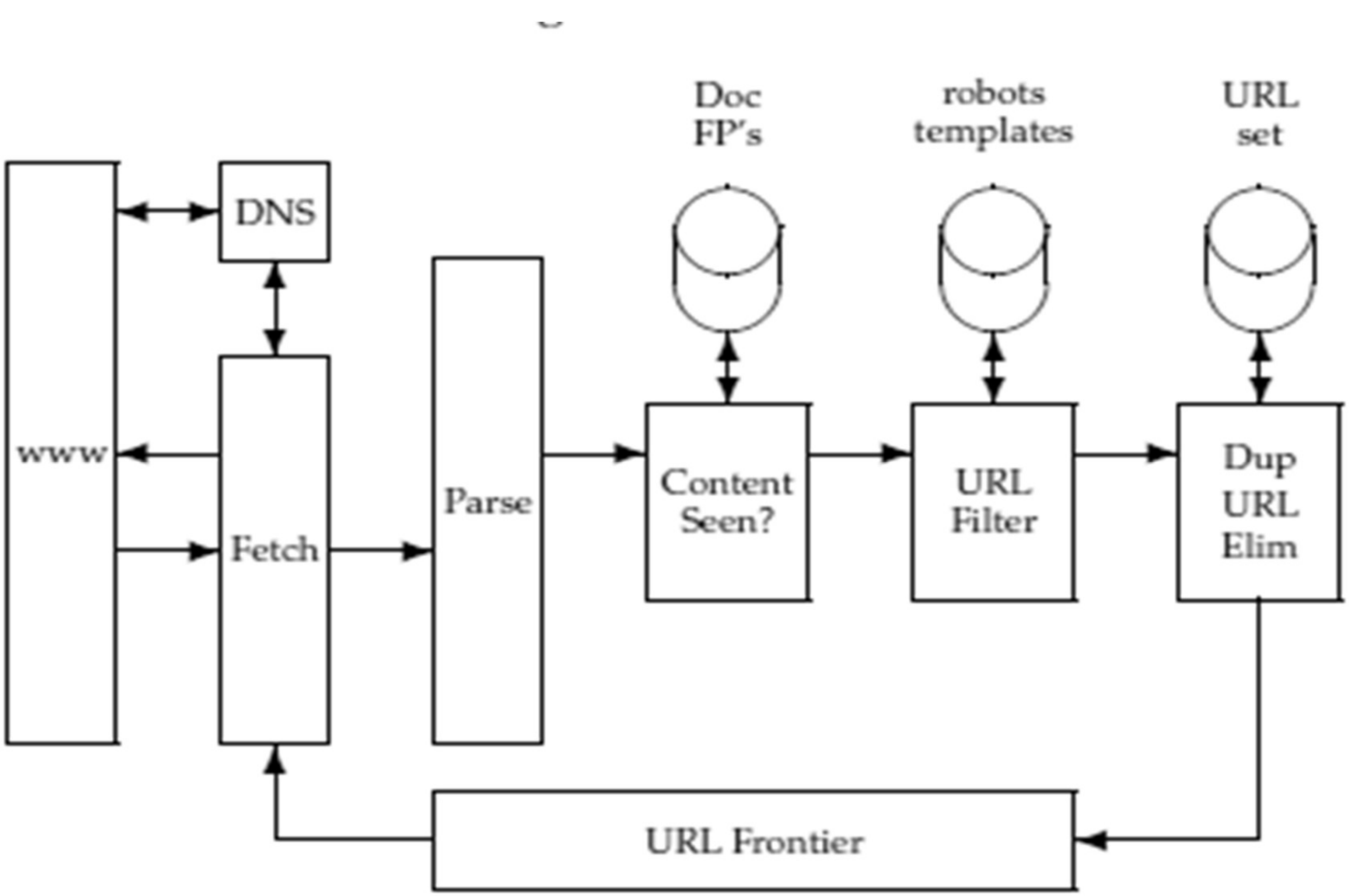
Architecture of Crawlers

- GoogleBot
- MsnBot /BingBot
- BaiduSpider
- Slurp by Yahoo
- Apache Nutch
- mnoGoSearch
- Dig
- GRUB

High-Level Architecture



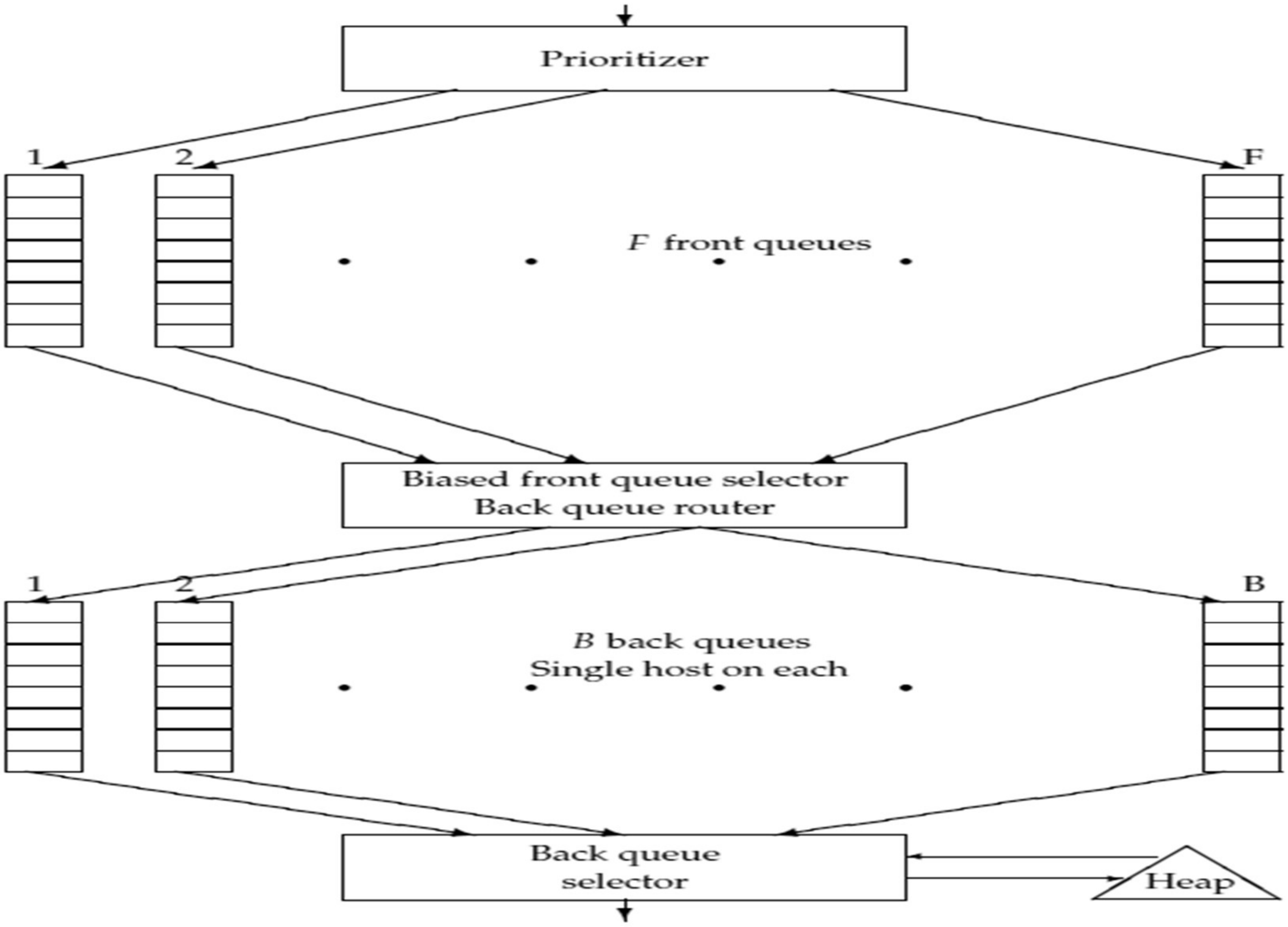
Architecture of a Crawler



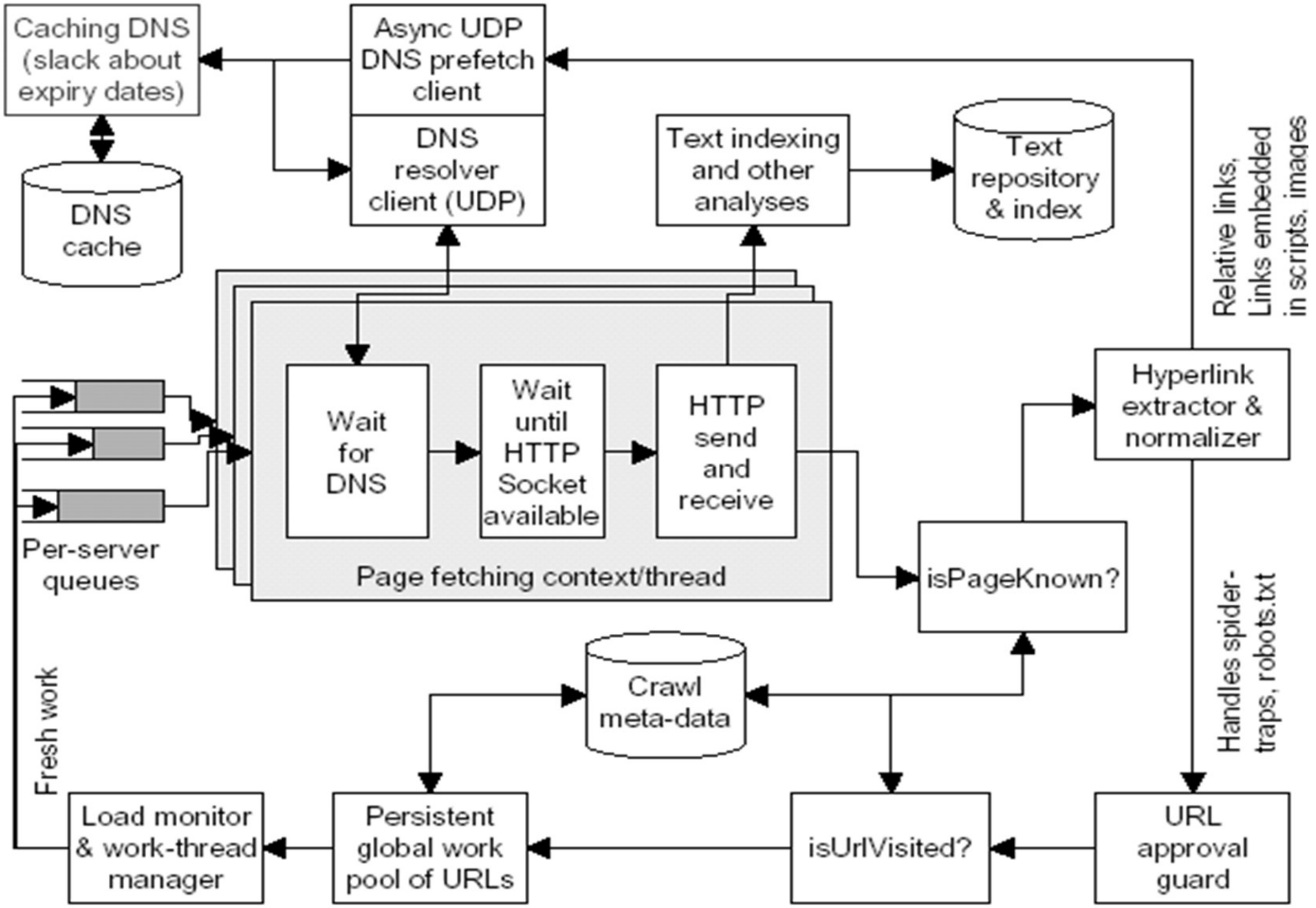
Architecture of a Crawler

- URL Frontier: containing URLs yet to be fetches in the current crawl. At first, a seed set is stored in URL Frontier, and a crawler begins by taking a URL from the seed set.
- DNS: domain name service resolution. Look up IP address for domain names.
- Fetch: generally use the http protocol to fetch the URL.
- Parse: the page is parsed. Texts (images, videos, and etc.) and Links are extracted.

URL frontier



Typical Crawler



Caching DNS (slack about expiry dates)
DNS cache
Async UDP DNS prefetch client
DNS resolver client (UDP)
Text indexing and other analyses
Text repository & index
Relative links, Links embedded in scripts, images
Wait for DNS
Wait until HTTP Socket available
HTTP send and receive
Page fetching context/thread
Per-server queues
Hyperlink extractor & normalizer
isPageKnown?
Handles spider-traps, robots.txt
Crawl meta-data
Fresh work
Load monitor & work-thread manager
Persistent global work pool of URLs
isUrlVisited?
URL approval guard