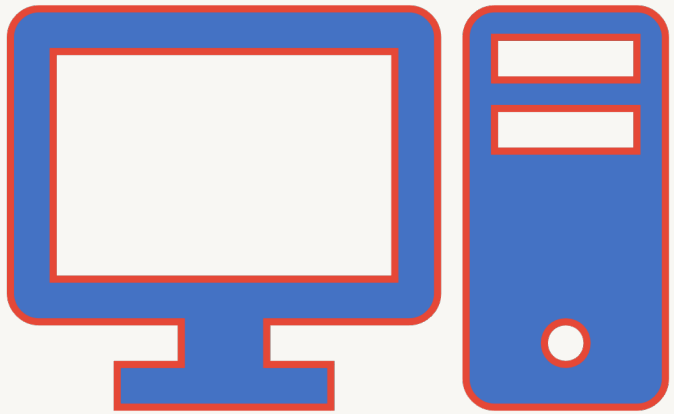
Web mining and content analysis

Crawling and indexing web content

By – **Shubham Gupta**(a1787223)
Siddhved Phadke(a1756264)
Group – **61**



Contents

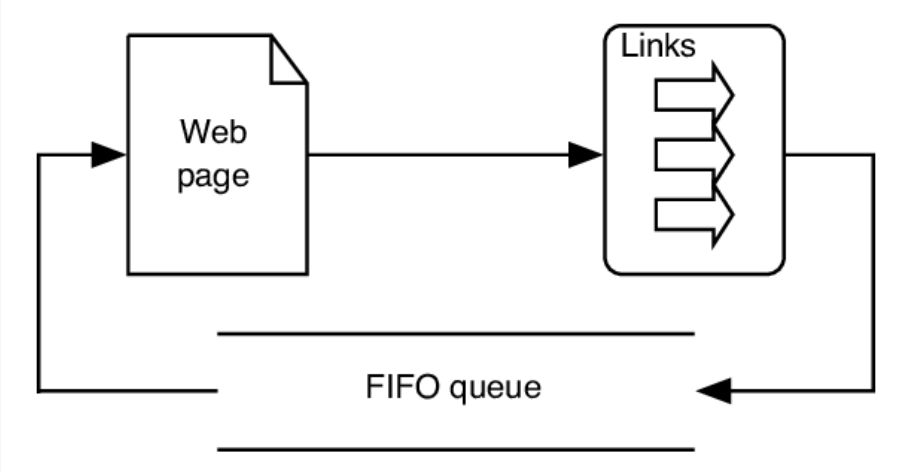
Introduction

- The web crawlers or “spiders” are small programs used by search engines to download pages.
- Then it will index the downloaded pages to provide fast search for later use.
- Web crawling can be done in automated or manual manner.
- Web crawlers are also known as web spiders and web robots.
- Used by most of the search engines such as Google, Yahoo, Bing, etc.

Types of Web Crawlers

- Breadth First Crawlers
- Incremental Web Crawlers
- Form Focused Web Crawler
- Focused Crawler
- Hidden Web Crawler
- Parallel Crawlers
- Distributed Web Crawler
- Distributed Focused Web Crawler

Breadth First Crawler[1]

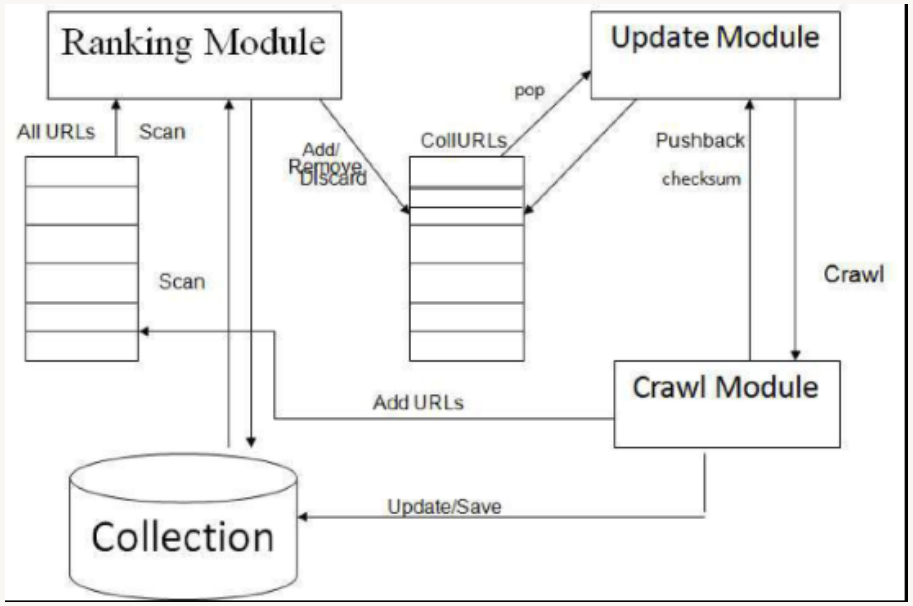


Breadth First Crawler

- Starts crawling around limited number of pages only.
- Follows breadth first algorithm.
- Decreases the average quality of pages which are downloaded.
- Decreasing the quality of the pages crawled increases the overall download rate and avoid overloading.

1: Vladislav Shkapenyuk and Torsten Suel. 2002. Design and implementation of a high-performance distributed web crawler. In Proceedings 18th

Incremental Web Crawler

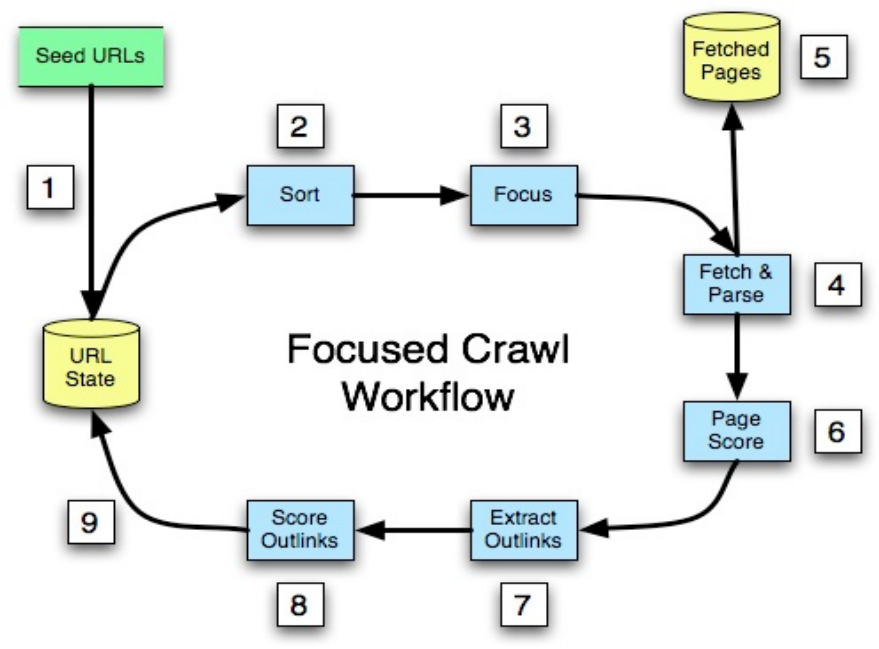


Incremental Web Crawler

- Usually crawler uses methodology of continually crawling the entire web.
- Instead of restarting the crawl on the web, this crawler crawls on the pages which are modified since crawl.
- Because of this nature, this crawler has achieved the low peak load performance and high quality of updated data.

András Nemeslaki and Károly Pocsarovszky. 2011. Web crawler research methodology. (2011).

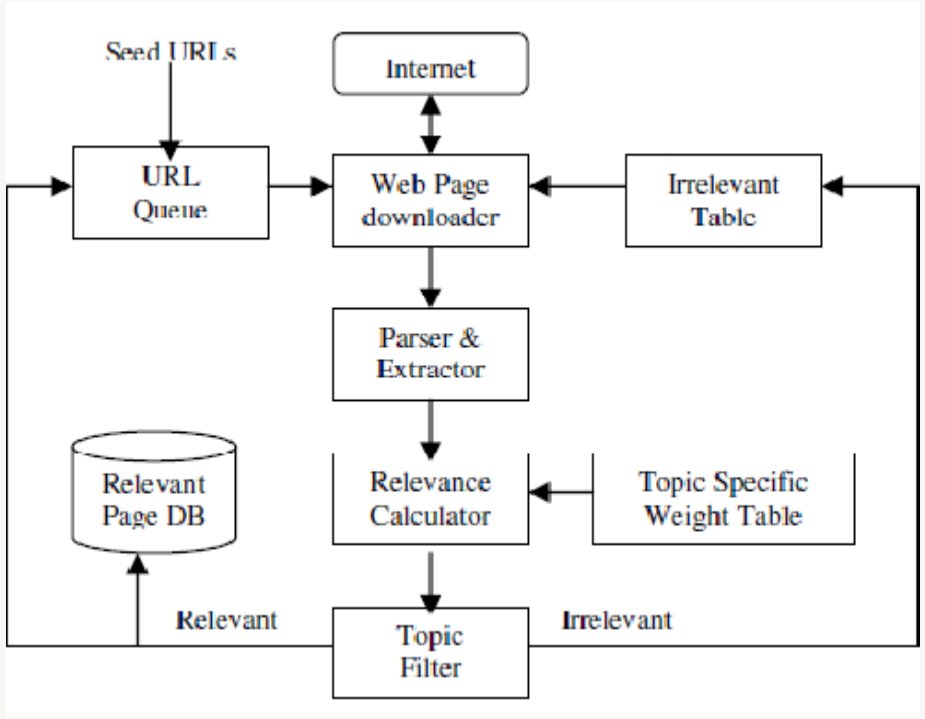
Form Focused Web Crawler



Form Focused Web Crawler

- Topic focused crawlers are developed to provide filters to irrelevant links while updating URL queue.
- Because of this nature, it achieves great reduction in search space with improved search efficiency and quality.
- However, it uses domain specific crawling it can miss forms of specific domain which resides on other relative pages.

Sandeep Sharma and Ravinder Guide Kumar. 2008. Web-Crawling Approaches in Search Engines. Ph.D. Dissertation.

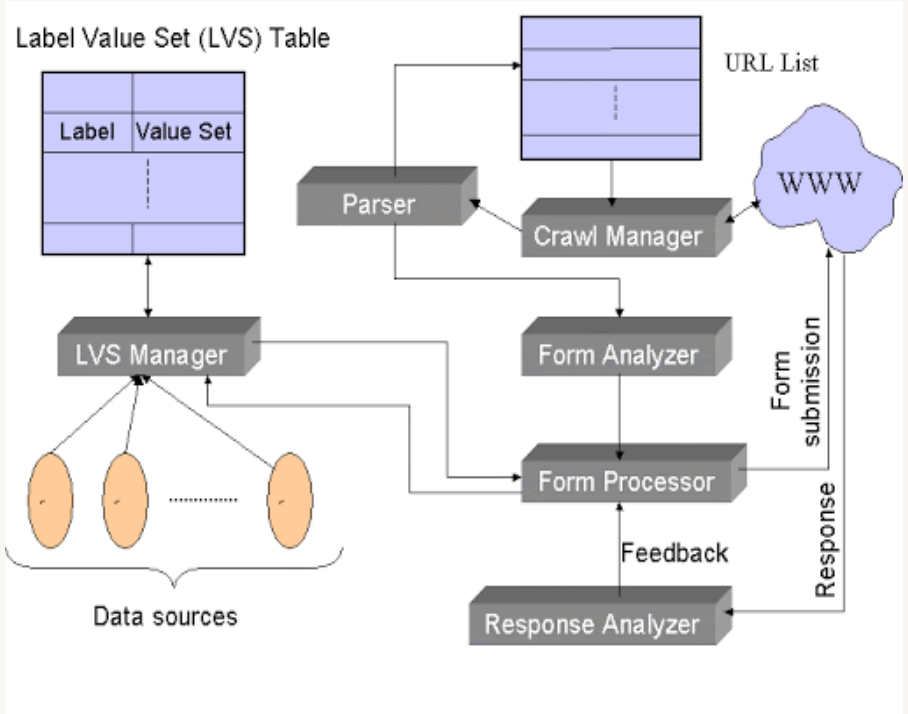


Focused Web Crawler

Focused Web Crawler

- Focused Web Crawler or hypertext classifier mainly composed of following three main components:
- ➢ Classifier used to make relevant decisions on pages to be crawled.
- ➢ Evaluate web site visit priorities.
- ➢ Reconfigures the priority controls for visiting web sites after every web site.

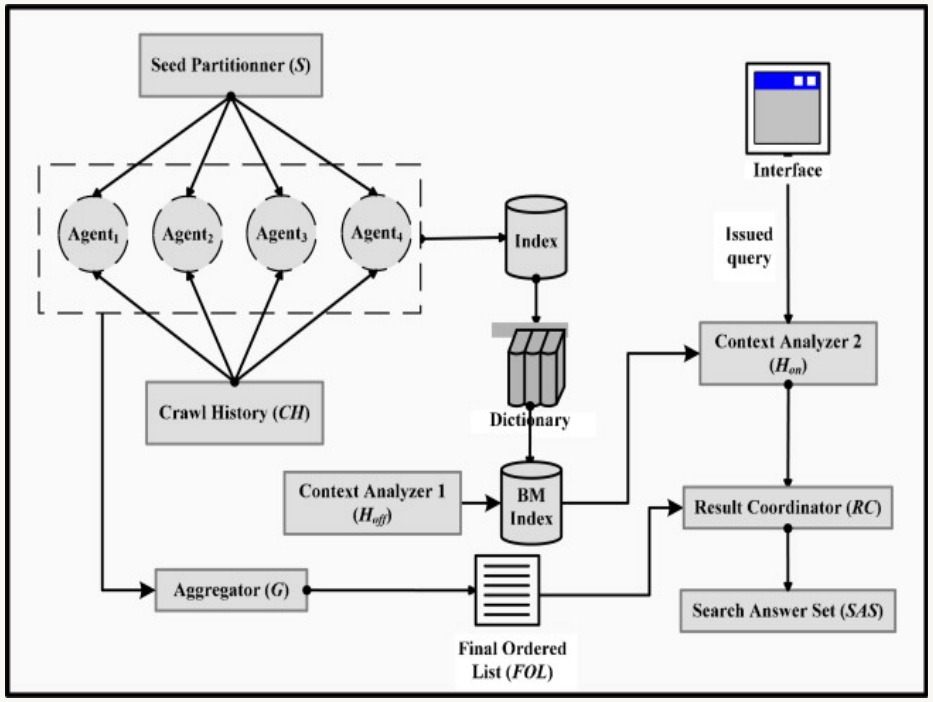
Soumen Chakrabarti, Martin Van den Berg, and Byron Dom. 1999. Focused crawling: a new approach to topic-specific Web resource discovery.



Hidden Web Crawler

Hidden Web Crawler

- Web based crawling based on database of the search engines.
- Can access information which is hidden behind search forms.
- Crawler that follows and crawls through web pages which requires user permission to access the data on web page.



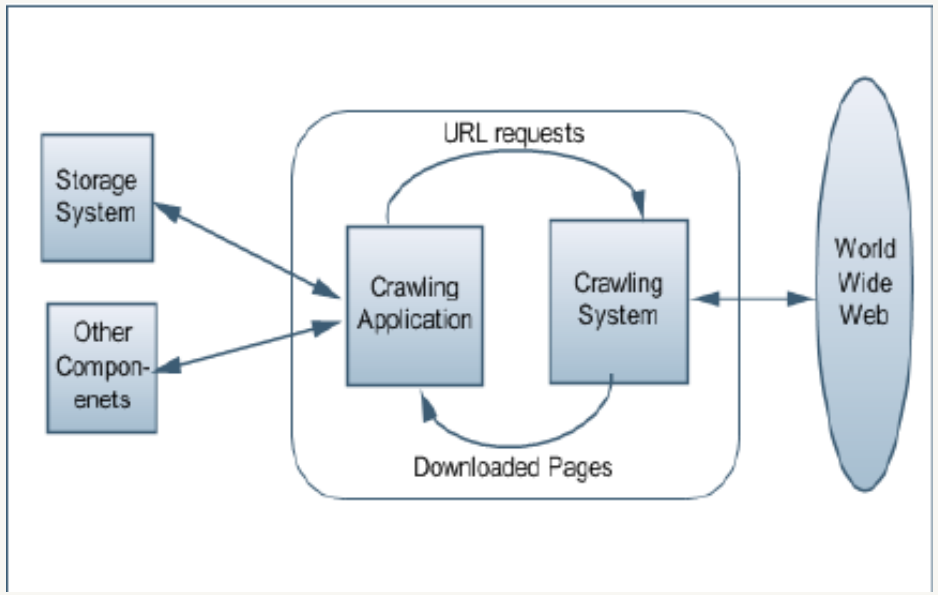
Parallel Web Crawler

Parallel Web Crawler

- Can work parallelly with other crawlers i.e. at the same time.
- Crawls over the distributed network of the web pages.
- Uses Random Access Memory of the systems to search through the history of the search and crawl related pages accordingly.

Junghoo Cho and Hector Garcia-Molina. 2002. Parallel crawlers. In Proceedings of the 11th international conference on World Wide Web. 124–135.

Distributed Web Crawler

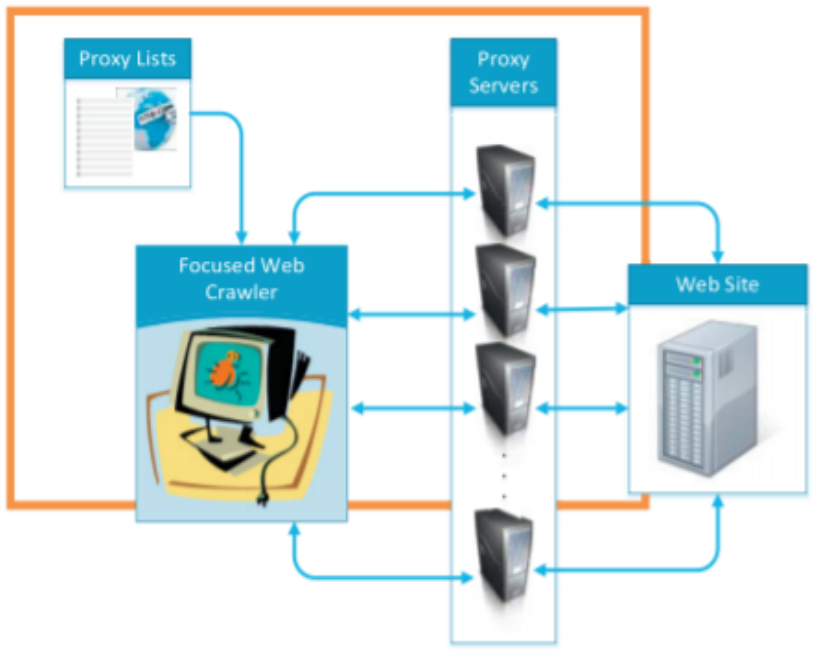


Distributed Web Crawler

- Runs on a network of workstations.
- Easier crawler to index the web pages.
- Efficient of cost reduction as it runs on workstations instead of running on single systems.
- Hardware requirements for the crawling process are significantly low than other crawlers.

Dhiraj Khurana and Satish Kumar. 2012. Web crawler: A review. International Journal of Computer Science & Management Studies 12, 1 (2012),

Distributed Focused Web Crawler



Distributed Focused Web Crawler

- Based on the database contents of the web pages.
- Supports multi threading architecture of crawler.
- Uses memory of the systems which is less efficient in real world.
- Due to this crawler data mining has significantly improved.

Harry T Yani Achsan and Wahyu Catur Wibowo. 2014. A fast distributed focused-web crawling. Procedia Engineering 69, 1 (2014), 492–499.

Web Crawling Architectures

MultiCrawler: A Pipelined Architecture for Crawling and Indexing Semantic Web Data[1]

1: Andreas Harth, Jürgen Umbrich, and Stefan Decker. 2006. Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In International Semantic Web Conference. Springer, 258–271.

Requirements of a Web Crawler

Performance and scalability –

- Keep up the constant growth taking place in the data source
- Manage data which is present on the web scale.
- Performance-oriented architecture
- Device should be able to scale up

Utilizing data from different formats –

- System needs to index data from multiple web sources.
- System must syntactically transform the indexed data and arrive at an integrated dataset.

1: Andreas Harth, Jürgen Umbrich, and Stefan Decker. 2006. Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In International Semantic Web Conference. Springer, 258–271.

Structure of a Multi-crawler[1]

- pipe-lined architecture which lead to a considerable performance improvements.
- Executing the different steps in parallel was the main idea
- The process of crawling was logically split into five phases.
- Namely Fetch, Detect, Transform, Index, and Extract as seen in Figure 1

Figure 1

1: Andreas Harth, Jürgen Umbrich, and Stefan Decker. 2006. Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In International Semantic Web Conference. Springer, 258–271.

Topical Web Crawlers: Evaluating Adaptive Algorithms[1]

- Encapsulate any specific crawling algorithm, the following system architecture was designed
- Any crawler can be directly plugged into the system through a general interface
- To optimize efficiency and at the same time not affecting evaluation

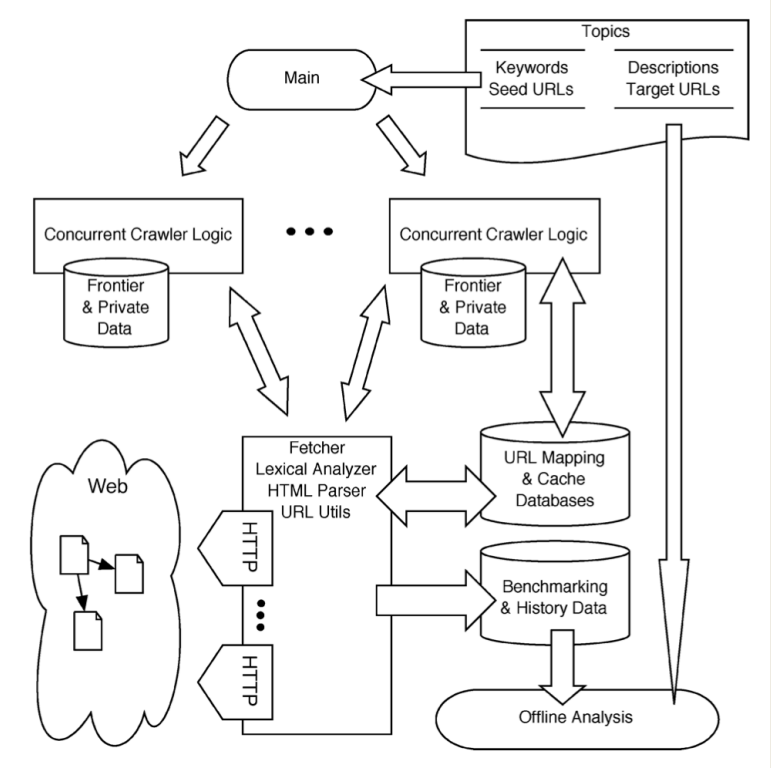


Figure 1

1: Filippo Menczer, Gautam Pant, and Padmini Srinivasan. 2004. Topical web crawlers: Evaluating adaptive algorithms. ACM Transactions on Internet Technology (TOIT) 4, 4 (2004), 378–419.

An Efficient Approach for Web Indexing of Big Data through Hyperlinks in Web Crawling[1]

- Uses an algorithmic process for Web Crawling
- Using depth first manner to crawl a web is the most effective way.
- Google's idea was one of the backdrop for the development of the proposed system.

1: R Suganya Devi, D Manjula, and RK Siddharth. 2015. An efficient approach for web indexing of big data through hyperlinks in web crawling. The Scientific World Journal 2015 (2015).

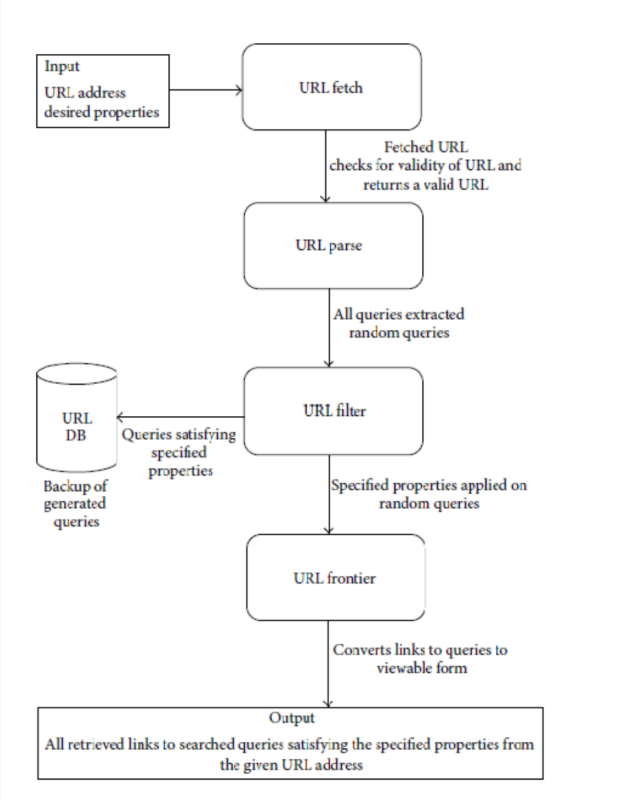


Figure 1

- We can see the entire proposed crawling architecture in the Figure 1.
- The **key features** of the proposed system are –

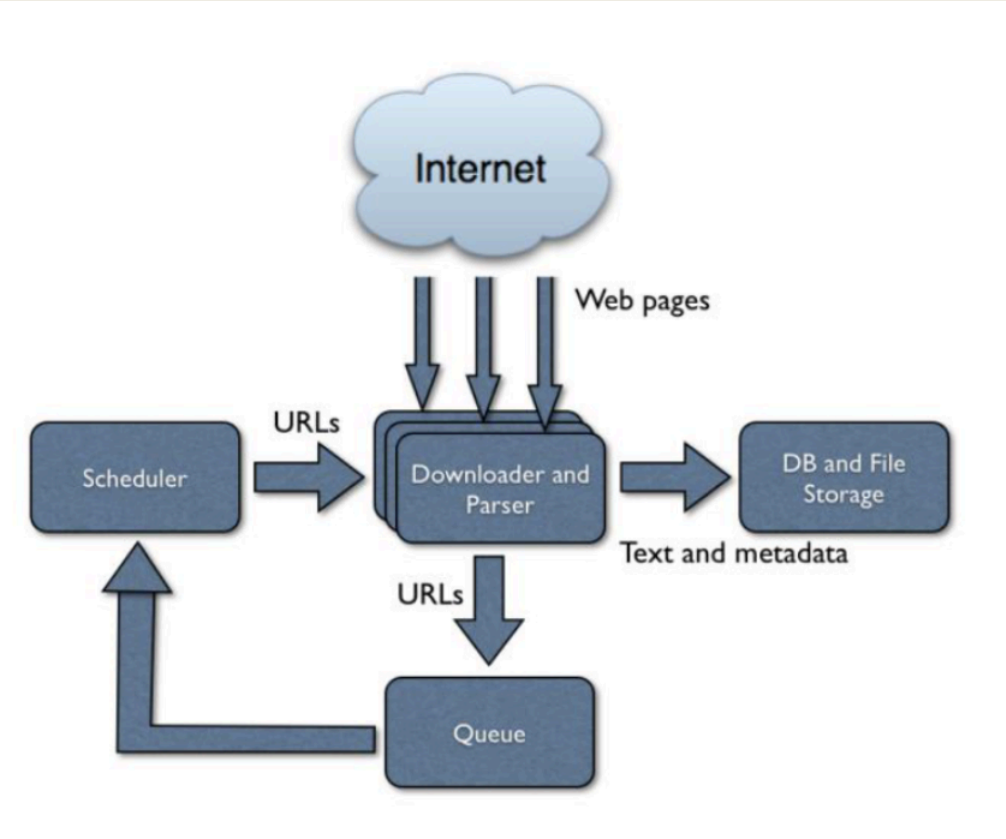
➢ Web needs to be identified and indexed.

➢ New pages should be be accessed from the old pages

➢ metatag content in the database should be stored

➢ Time-consuming calculations such as Page-Rank should be always avoided.

Web Crawler: Extracting the Web Data[1]



Web Crawler Architecture

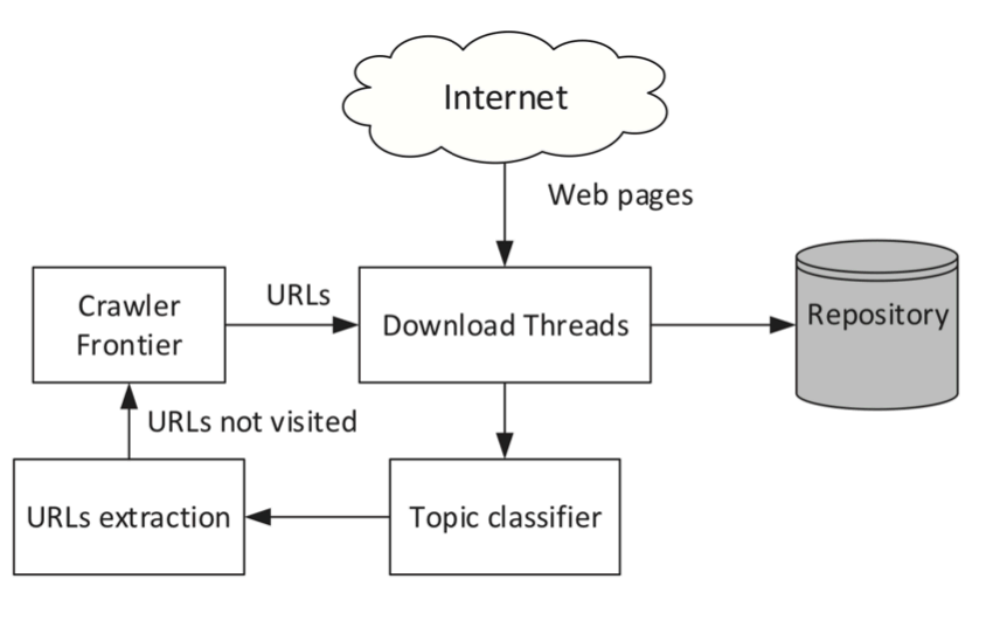
- Each one of the available URL is called a "seed"
- The crawler visits each one of the listed URLs.
- All the hyperlinks on the website are detected by the web crawler and are added to the list of the URLs to visit subsequently.
- There are policies and rules in place for the URLs that needs to be visited.
- Now according to the requirement, all the related file and links being searched can be made available.

1: Niraj Singhal, RP Agarwal, Ashutosh Dixit, and AK Sharma. 2011. Information retrieval from the web and application of migrating crawler. In 2011 International Conference on Computational Intelligence and Communication Networks. IEEE, 476–480.

A Semantic Focused Web Crawler Based on a Knowledge Representation Schema[1]

Composed of three main components:

- **URL Frontier**
- **Page Downloader**
- **repository**



Focused Web Crawler

1: Julio Hernandez, Heidy M Marin-Castro, et al. 2020. A Semantic Focused Web Crawler Based on a Knowledge Representation Schema. Applied Sciences 10, 11 (2020), 3837.

- Machine Learning Classification algorithms is one of the technique on which a focused web crawler is based on
- To find relevant web pages and at the same time adding it to the local database[1]
- expressiveness of the content in the web pages are not considered in such approaches.

1: Julio Hernandez, Heidy M Marin-Castro, et al. 2020. A Semantic Focused Web Crawler Based on a Knowledge Representation Schema. Applied Sciences 10, 11 (2020), 3837.

Web Crawling Algorithms

We aim to describe and evaluate **five different crawling algorithms**:

- Breadth-First
- Best-First
- PageRank
- Shark-Search
- Info-Spiders

Breadth-First

- A Breadth-First crawler is the easiest technique for Crawling
- It utilises the frontier as a queue for FIFO, crawling links in the the order they are in.
- Breadth-First is used as a baseline crawler to give a lower bound to more sophisticated algorithms.
- It is used as a baseline because no awareness of the subject is used in it.

```
Breadth-First (starting_urls) {
   foreach link (starting_urls) {
      enqueue(frontier, link);
   }
   while (visited < MAX_PAGES) {
      link := dequeue_link(frontier);
      doc := fetch(link);
      enqueue(frontier, extract_links(doc));
      if (#frontier > MAX_BUFFER) {
         dequeue_last_links(frontier);
      }
   }
}
```

Breadth First Crawler

Best-First

- Best link is selected for crawling given a boundary of links.
- Various Best-First strategies of increasing complexity, efficacy could be developed based on sophisticated link estimation parameters.

```
BFS (topic, starting_urls) {
   foreach link (starting_urls) {
      enqueue(frontier, link, 1);
   }
   while (visited < MAX_PAGES) {
      link := dequeue_top_link(frontier);
      doc := fetch(link);
      score :=  sim(topic, doc);
      enqueue(frontier, extract_links(doc), score);
      if (#frontier > MAX_BUFFER) {
         dequeue_bottom_links(frontier);
      }
   }
}
```

Pseudocode

PageRank

- PageRank[1] was proposed as a possible model of user surfing behaviour.
- reflects the possibility of a random surfer, being on that page at any given time.
- Score of a page is recursively based on the scores of the pages that point to it.
- PageRank was meant to be used in tandem with the ranking of recovered sets of documents with content-based criteria.
- This is in fact how PageRank is used in the Google search engine.

```
PageRank (topic, starting_urls, frequency) {
   foreach link (starting_urls) {
      enqueue(frontier, link);
   }
   while (visited < MAX_PAGES) {
      if (multiplies(visited, frequency)) {
         recompute_scores_PR;
      }
      link := dequeue_top_link(frontier);
      doc := fetch(link);
      score_sim :=  sim(topic, doc);
      enqueue(buffered_pages, doc, score_sim);
      if (#buffered_pages >= MAX_BUFFER) {
         dequeue_bottom_links(buffered_pages);
      }
      merge(frontier, extract_links(doc), score_PR);
      if (#frontier > MAX_BUFFER) {
         dequeue_bottom_links(frontier);
      }
   }
}
```

Pseudocode of Page Rank

1: Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. (1998).

- PageRank requires a recursive calculation, so its computation can be a very resource-intensive process before convergence.
- To improve efficiency and computational overheads it is recomputed only at regular intervals.
- PageRank crawler can be seen as a variant of Best-First.

Shark-Search[1]

- The algorithm stops looking in regions which do not generate appropriate pages.
- Shark-Search offers mainly two improvements-

➢ Uses a continuously valued function to calculate relevance.

➢ Refined definition of possible scores for links at the frontier of the crawl.

- more sophisticated link assessment feature, the Shark-Search crawler can be seen as a variant of Best-First.

1: Michael Hersovici, Michal Jacovi, Yoelle S Maarek, Dan Pelleg, Menachem Shtalhaim, and Sigalit Ur. 1998. The shark-search algorithm. An application: tailored Web site mapping. Computer Networks and ISDN Systems 30, 1 (1998), 317–326.

Pseudocode of Shark Search Crawler

```
Shark (topic, starting_urls) {
   foreach link (starting_urls) {
      set_depth(link, d);
      enqueue(frontier, link);
   }
   while (visited < MAX_PAGES) {
      link := dequeue_top_link(frontier);
      doc := fetch(link);
      doc_score :=  sim(topic, doc);
      if (depth(link) > 0) {
         foreach outlink (extract_links(doc)) {
            score = (1-r) * neighborhood_score(outlink)
                  +  r   * inherited_score(outlink);
            if (doc_score > 0) {
               set_depth(outlink, d);
            } else {
               set_depth(outlink, depth(link) - 1);
            }
            enqueue(frontier, outlink, score);
         }
         if (#frontier > MAX_BUFFER) {
            dequeue_bottom_link(frontier);
         }
      }
   }
}
```

Info-Spiders

- An adaptive agent population uses evolving query vectors and neural nets to scan for pages related to the subject to determine which links to pursue.
- This approach utilises a fitness measure dependent on similarity as a criterion for local selection.
- Due to its extreme local behaviour and the lack of any memory, this crawler was outperformed by BFS.

1: Filippo Menczer, Gautam Pant, and Padmini Srinivasan. 2004. Topical web crawlers: Evaluating adaptive algorithms. ACM Transactions on Internet Technology (TOIT) 4, 4 (2004), 378–419.

- Since then, several improvements have been made to the original algorithm while retaining its ability to learn link estimates.
- Achieved via neural networks and focus its search through selective reproduction towards more promising areas.
- Info-Spiders have the unique ability to adjust the link-following behaviour during a crawl
- Info-Spiders was the first crawling algorithm to use reinforcement learning

```
IS (topic, starting_urls) {
   agent.topic := topic;
   agent.energy := 0;
   agent.fsize := MAX_BUFFER;
   agent.frontier := starting_urls;
   insert(agent, population);
   while (visited < MAX_PAGES) {
      foreach parallel agent (population) {
         link := pick_and_dequeue(agent.frontier);
         doc := fetch(link);
         newenergy = sim(agent.topic, doc);
         agent.energy += newenergy;
         learn_to_predict(agent.nnet, newenergy);
         foreach outlink (extract_links(doc)) {
            score = ALPHA * newenergy +
                    (1 - ALPHA) * agent.nnet(doc, outlink);
            enqueue(agent.frontier, outlink, score);
         }
         if (#agent.frontier > agent.fsize) {
             dequeue_bottom_links(agent.frontier);
         }
         delta := newenergy - sim(topic, agent.doc);
         if (boltzmann(delta)) {
            agent.doc := doc;
         }
         if (agent.energy > THETA and pop_size < MAX_POP_SIZE) {
            child.topic := expand(agent.topic, agent.doc);
            (child.energy, agent.energy) := (0, 0);
            (child.fsize, agent.fsize) := half(agent.fsize);
            (child.frontier, agent.frontier) := split(agent.frontier);
            child.nnet := extend(agent.nnet, child.topic)
            insert(child, population);
         }
      }
   }
}
```

Challenges

- No headers found for URL based web crawler.
- Wrong dates headers forging URL based web crawler.
- Detection of unwanted blogs, forums and irrelevant links.
- Detecting duplicated content on several web pages.
- Using cache memory of systems or workstations leads to slow and erroneous web pages to crawl.

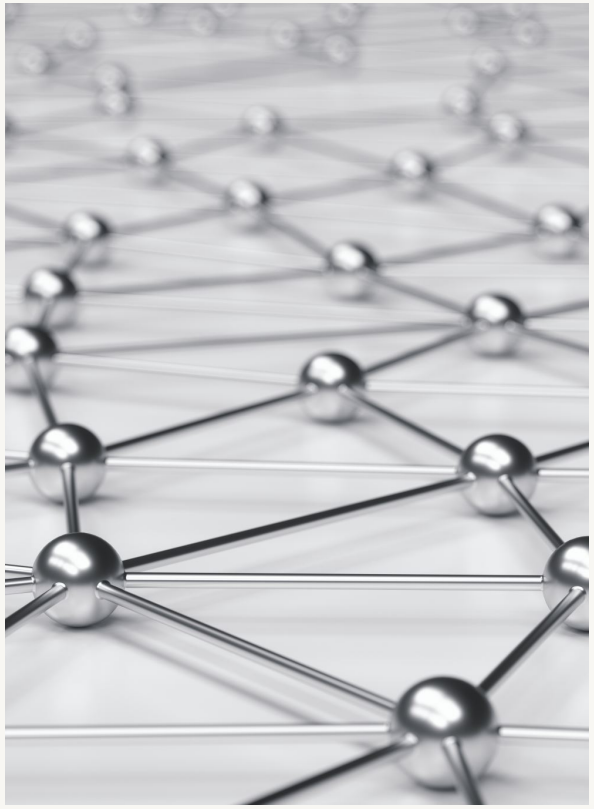
Future Work

Multi-crawler pipe-lined architecture

- This architecture leads to bottleneck of URL queue as it follows multiple threads at once.
- Future work for this architecture can be achieved by increasing the linkage between source of web content with system using RDF(Resource Description Framework).
- Can be tested for longer times on large number of datasets to test the efficiency.

Topical Web Crawlers

- This architecture can be said reliable when it comes crawling on the small number of data.
- Future work for this web crawler can be achieved by increasing great efficiency using other algorithms rather than using InfoSpider algorithm.
- Efficiency and scalability of this crawler can be improved by improving quality of the hardware used in the system and also increasing the bandwidth of network.



Semantic Focused Web Crawler

- It usually focuses on the concept of the Knowledge Representation Schema(KRS).
- Future work consists of reducing the bandwidth of the web pages as this architecture builds its own database on search engine server.
- Also future work consists of developing new approach for KRS construction so that crawling of the web pages can be improved.

Conclusion

In this presentation our aim was to –

- Study and evaluate the different types of architectures and systems, As web crawlers are one or if not the most important aspect of all search engines.
- We found some gaps in the research and proposed some future work which could help build better web crawling and indexing systems.

YOUTUBE VIDEO LINK

https://youtu.be/OXErWCuaS-A