
Chetna, Harpal Tanwar, Navdeep Bohra

ABSTRACT: Now days search engine transfers the web data from one place to another. They work on client server architecture where the central server manages all the information. A web crawler is a program that extracts the information over the web and sends it to the search engine for further processing. It is found that maximum traffic (approximately 40.1%) is due to the web crawler. The proposed scheme shows how web crawler can reduce the traffic using Dynamic web page and HTTP GET request using asp.net.

KEYWORDS: Crawler, Search Engine, WWW.

I. INTRODUCTION

All the search engines have powerful crawlers that visit the internet time to time for extracting the useful information over the internet. The retrieved pages are indexed and stored in the data base as shown in figure 1. Actually Internet is a directed graph, or web page as a node and hyperlink as edge, so the search operation could be abstracted as a process of traversing directed structure graph. By following the linked structure of the web, we can traverse a number of new pages started from starting web pages. Web crawlers are designed to retrieve web pages and add them their represent to the local repository/databases [1]. Crawler updates their information once a week, sometimes it update monthly or quarterly also. They cannot provide up-to-date version of frequently updated pages. To catch up frequent updates without putting a large burden on content provider, we believe retrieving and processing data near the data source is inevitable [2]. Currently more than one search engines are available in the market. That increase in complexity of web traffic has required that we base our model on the notation of web request rather than the web pages.

Web crawler are software systems that use the text and links on web pages to create search indexes of the pages, using HTML links to follow or crawl the connections between pages.

Manuscript Received on July 2012.
Chetna, M.TECH (Scholar) Jagannath University, Jaipur (Rajasthan), India
Harpal Tanwar, Lecturer, BMCTM, Farukhnagar (MDU), India
Navdeep Bohra, Assistant Professor, MSIT, Janakpuri, New Delhi, India

Figure 1 Architecture of a web search engine [5]

The WWW is a web of hyperlinked repository of trillions of hypertext documents [9] laying on different web sites. World Wide Web (Web) traffic continues to increase and is now estimated to be more than 70 percent of the total traffic on the Internet [3].

A. Basic Crawling Terminology

We need to know some basic terminology of web crawler which plays an important role in implementation of the web crawler.

Seed page: Crawling means to traverse the web recursively by picked up the starting URL from the set of URL. Starting URL is entry point from where all the crawlers start their searching procedure. This set of URL known as seed page.

Frontier: The crawling procedure starts with a given URL. Extracting the link from it and adding them to an unvisited list of URL. this unvisited list known as frontier. The frontier implemented by a queue.

Parser Parsing may imply simple hyperlinked/URL extraction or it may involve the more complex process of tidying up the HTML content in order to analyze the HTML tag tree. The job any parser is to parse the fetched pages to extract the list of new URL from it and return the
new unvisited URL to the frontier.

The Basic algorithm of a web crawler is given below:

Start
Read the URL from the seed URL.
Check whether the documents already downloaded or not
If documents are already downloaded.
   Break.
Else
   Add it to the frontier.
Now pick the URL from that frontier and extract the new link from it. Add all the newly found URL into the frontier.
Continue.
End

The main function of a crawler is to add new links into the frontier add to select a new

II. RELATED WORK

To reduce the web crawler traffic many researchers has completed their research in following areas:

- In this author used dynamic web pages with HTTP Get request with last visit parameter [4].
- One approach is the use of active network to reduce unnecessary crawler traffic [6].
- The author proposed an approach which uses the bandwidth control system in order to reduce the web crawler traffic over the internet [7].
- One is to place the mobile crawler at web server. Crawler check updates in web site and send them to the search engine for indexing [8].
- Design a new web crawler using VB.NET technology [9].

III. PERFORMANCE MATRICES

In the implementation of web crawler we have taken some assumptions into the account just for simplifying algorithm and implementation and results.

- Remove a URL from the URL list
- Determine the protocol of underlying host like http, ftp etc.
- Download the corresponding document.
- Extract any links contained in it.
- Add these links back to the URL list.

IV. SIMULATOR

The simulator has been designed to study the behavior pattern of different crawling algorithms from the same set of URLs. We designed a crawler using VB.NET and ASP.NET window application project type our crawler can work on globally and locally, means it can give result on intranet and internet. It use URL in a format like http://www.yahoo.com and set a location or name for saving crawling results data in MS Access database.
millisecond time of system, return by C#, millisecond time is maintained by “update” data structure. First we perform crawling on web site using old approach. Then we perform crawling using proposed approach. When We perform the web crawling on website. The results obtained shown in Table 1.

**TABLE 1**

<table>
<thead>
<tr>
<th>Index</th>
<th>URL</th>
<th>Start Time</th>
<th>End Time</th>
<th>Time to Reach this URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://localhost:64079/Crawlerbychetna/index.html">http://localhost:64079/Crawlerbychetna/index.html</a></td>
<td>1337504704175</td>
<td>1337504704731</td>
<td>556</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://localhost:64079/Crawlerbychetna/About.html">http://localhost:64079/Crawlerbychetna/About.html</a></td>
<td>1337504704175</td>
<td>1337504704798</td>
<td>603</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://localhost:64079/Crawlerbychetna/branch.html">http://localhost:64079/Crawlerbychetna/branch.html</a></td>
<td>1337504704175</td>
<td>1337504704893</td>
<td>718</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://localhost:64079/Crawlerbychetna/contact.html">http://localhost:64079/Crawlerbychetna/contact.html</a></td>
<td>1337504704175</td>
<td>1337504705011</td>
<td>836</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://localhost:64079/Crawlerbychetna/service.html">http://localhost:64079/Crawlerbychetna/service.html</a></td>
<td>1337504704175</td>
<td>1337504705118</td>
<td>1006</td>
</tr>
<tr>
<td>6</td>
<td><a href="http://localhost:64079/Crawlerbychetna/Person.html">http://localhost:64079/Crawlerbychetna/Person.html</a></td>
<td>1337504704175</td>
<td>1337504705145</td>
<td>1170</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://localhost:64079/Crawlerbychetna/nquery.html">http://localhost:64079/Crawlerbychetna/nquery.html</a></td>
<td>1337504704175</td>
<td>1337504705160</td>
<td>1385</td>
</tr>
</tbody>
</table>

To test the proposed approach we direct the web crawler to dynamic web page dynamic.aspx and set the last visit time at URL and perform crawling.

**Test 1:** Update time and URL of pages index, branch and person in “Update” data structure at web crawler set the LAST_VISIT time before time of pages in the Update. Performed crawling, results obtained are shown in table 2.

**TABLE 2**

<table>
<thead>
<tr>
<th>Index</th>
<th>URL</th>
<th>Start Time of Crawler (Time in millisecond)</th>
<th>End Time of Crawler (Time in millisecond)</th>
<th>Time to Reach this URL (Time in millisecond)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><a href="http://localhost:64079/Crawlerbychetna/index.html">http://localhost:64079/Crawlerbychetna/index.html</a></td>
<td>1337504704732</td>
<td>1337504705678</td>
<td>293</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://localhost:64079/Crawlerbychetna/branch.html">http://localhost:64079/Crawlerbychetna/branch.html</a></td>
<td>1337504704732</td>
<td>1337504705336</td>
<td>396</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://localhost:64079/Crawlerbychetna/Person.html">http://localhost:64079/Crawlerbychetna/Person.html</a></td>
<td>1337504704732</td>
<td>1337504705119</td>
<td>531</td>
</tr>
</tbody>
</table>

**Test 2:** Update time and URL of page about in “Update” data structure. At web crawler sets the LAST_VISIT time, before the time of pages in the update. Performed crawling, results obtained are shown in table 3.

### TABLE 3

<table>
<thead>
<tr>
<th>Index</th>
<th>URL</th>
<th>Start Time</th>
<th>End Time</th>
<th>Time to Reach this URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><a href="http://localhost:64079/Crawlerbychetna/About.html">http://localhost:64079/Crawlerbychetna/About.html</a></td>
<td>1337511874 567</td>
<td>133751187497 2</td>
<td>290</td>
</tr>
</tbody>
</table>

**Test 3:** Update time and URL of pages service and query in “Update” data structure. At web crawler set the LAST_VISIT time before time of pages in the Update. Performed crawling, results obtained are shown in table 4.

### TABLE 4

<table>
<thead>
<tr>
<th>Index</th>
<th>URL</th>
<th>Start Time of Crawler (Time in millisecond)</th>
<th>End Time of Crawler (Time in millisecond)</th>
<th>Time to Reach this URL (Time in millisecond)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><a href="http://localhost:64079/Crawlerbychetna/service.html">http://localhost:64079/Crawlerbychetna/service.html</a></td>
<td>133751188786 1</td>
<td>1337511888753</td>
<td>892</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://localhost:64079/Crawlerbychetna/nquery.html">http://localhost:64079/Crawlerbychetna/nquery.html</a></td>
<td>133751188786 1</td>
<td>1337511888679</td>
<td>450</td>
</tr>
</tbody>
</table>

**B. Result**

In normal crawling is a time consuming process because crawler visit every web page to know all updated information in web site. In normal crawling it visits a total of 7 pages. Crawler takes 1385 milliseconds to visit complete site. In proposed approach crawler visits Dynamic update page and updated web pages only. Crawler take about 500 milliseconds when there are 3 updates, about 450 milliseconds when there are two update. When there are three updates in experimental Website proposed sachem is 4.83 time faster than old approach. With two updates proposed scheme is 7.03 times faster than old scheme.

Graph 1 shows time taken by web crawler to download updates. In normal crawling crawler visits 7 pages to find updates. But number of page visit is very small in proposed approach. When there is one update crawler only visit 2 pages and when there are 2 updates crawler only visits 3 pages. If there are 3 updates in web site crawler visit 4 pages.
V. CONCLUSION

With this approach Crawler find new updates on the web server using Dynamic web page. Using this crawler you can send the queries with requested URLs and can reduce the maximum crawler traffic over the internet. It is found that approximately 40.1% traffic is due to the web crawler. So that using this method you can reduce 50% traffic of the web crawler (means half of the web crawler traffic i.e. 20% over the internet). The future work of this paper will be we can reduce the crawler traffic using page rank method and by using some parameters like as last modified parameter. This parameter tells the modified date and time of the fetched page. Last modified parameter can be used by the crawler for fetching the fresh pages from the Websites.

REFERENCES:


