International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 4 Ver. IV|| April 2018 || PP 42-50

An Appraisal of Web Crawling Algorithms

Oni O. A. & Chinedu Ukeje
Department Of Computer Studies, The Polytechnic, Ibadan, Oyo State Nigeria

Abstract :*Central To Any Research Work Or Data-Mining Project Is Having Sufficient Amounts Of Data That Can Be Processed To Provide Meaningful And Statistically Relevant Information. It Is Almost Impossible For Anybody Using The World Wide Web To Get Meaningful Information Without Having To Use A Search Engine. In This Study, We Appraisal Different Crawling Algorithms; Their Strength And Weakness And Also Present The Ideology Behind The Development Of These Algorithms. We Then Go A Step Further To Discuss The Architecture Of A Web Crawler And The General Crawling Policy Of Web Crawlers.*
Keywords– *Web, Internet, Crawler, Search Engine, Crawling Algorithm.*

Date of Submission: 05-03-2018 Date of acceptance: 20-04-2018

I. Introduction

Since The Introduction Of The Internet, The Movement Of Information And The Flow Of Knowledge Have Greatly Increased. The Volume Of Data On The Internet Is Staggering And The Fact That This Information Or Data Is Increasing Every Day Is Also Mind Bending. The Average Internet Surfer Would At One Time Or The Other Need Specific Information Regarding A Particular Topic; This Could Prove Very Difficult (Without A Search Engine Like Google) As The Surfer Would Be Required To Know The Exact Website On Which To Get That Information. The Need To Provide Solution To Problems Like This Gave Birth To The Web Crawler.

A Web Crawler Is An Internet Bot That Systematically Browses The World Wide Web, Typically For The Purpose Of Web Indexing. A Web Crawler May Also Be Called A Web Spider, An Ant, An Automatic Indexer, Or A Web Scutter. Web Crawlers Are Programs Which Traverse Through The Web Searching For Relevant Information Using Algorithms That Narrow Down The Search By Finding Out Information That Is Similar And Relevant To The Search Term. This Process Is Iterative, As Long As The Results Are Relevant To The Surfer's Search Term.

The Crawler Algorithm Determines The Relevance Of The Result To The Search Term Based On Factors Such As Frequency And Location Of Keywords. Crawlers Gained Their Name Because Of The Fact That They Crawl Websites One Page At A Time As Would A Crawling Insect. Thy Follow The Pages From One Site To Another Until They Finish The Crawling Process.

Here Is The Process That A Web Crawler Follows:

1. Start From One Preselected Page. We Call The Starting Page The "Seed" Page.
2. Extract All The Links On That Page. (This Is The Part We Will Work On In This Unit And Unit 2.)
3. Follow Each Of Those Links To Find New Pages.
4. Extract All The Links From All Of The New Pages Found. Follow Each Of Those Links To Find New Pages.
5. Extract All The Links From All Of The New Pages Found.

Basically There Are Four Types Of Crawlers:

A) Traditional Crawler – Visits Entire Web And Replaces Index
B) Periodic Crawler – Visits Portions Of The Web And Updates Subset Of Index
C) Incremental Crawler – Selectively Searches The Web And Incrementally Modifies Index
D) Focused Crawler – Visits Pages Related To A Particular Subject

II. Webcrawling Fundamentals

From Mathematical Point Of View, The Internet Can Be Seen As A Large Directed Graph, Where Nodes Are Resources And Connections Between Them – Links, Otherwise Known As Urls(Uniform Resource Locator). This Graph Can Also Be Seen As A Tree Data Structure By Using Domain Name System And File System As Media, Where Sub-Domains And Sub-Directories Are Considered As Different Levels Of This Tree. This Forms A Kind Of File System That Defines Particular Path Structures.

Crawling Is A Process Of Mapping Dynamic Graph And Storing Node Data. This Process Is Done By Examining A Node, Finding Edges, And Repeating The Process With Nodes That Are Connected To This

Node. This Process Cycle Is Infinite If No Tracking Is Made Or If Process Has History Of Viewed Resources, But Has Revisit Policy As Well.

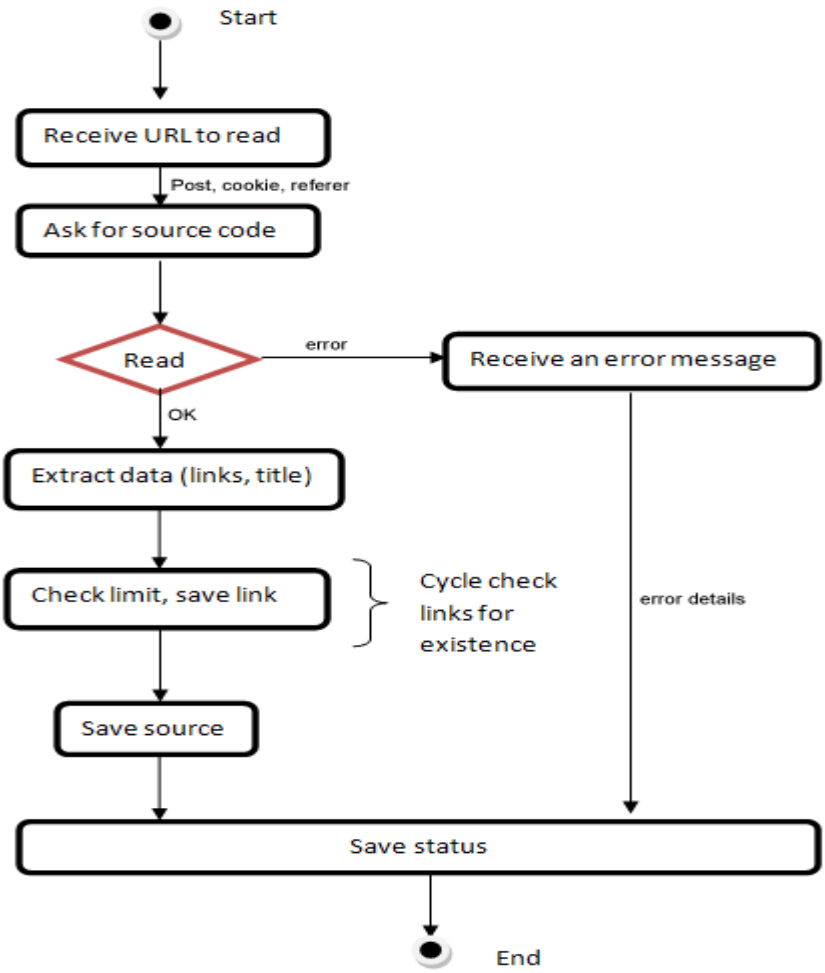

Figure 1. Crawling Cycle Functional Diagram

For Selection Policy And Search Engine Result Ordering The Graph Must Be Ordered. This Can Be Done By Analyzing General Graph Structure, Assigning Order With Human Aid, Adding Knowledge Discovery Algorithms, Tracking Popularity Of Web-Sites. Some Of Them Are Explained Further.

2.1 Page Ranking Algorithms

Ranking A Web-Page Or Entire Domain Is Needed To Set A Value Of Its Importance. Automatic Ranking Cannot Do What Human Does, Since It Has No Real Interest Or Field Of Research And That Is Why Importance Is Evaluated As Popularity.

In Global Scale, Popularity Cannot Be Measured By Making A Counter Of Visits For Every Page, Because This Cannot Be Solved Technically Without Proper Agreement With Web-Site Owner. Although Many Counter Services Have Been Offered, They Are All Voluntary. The Major Disadvantage Is Sharing Of Web-Site Privacy Policy With The Statistics Service.

Ways To Analyze Popularity Based On Indirect Facts Are Used In Modern Search-Engines.

2.1.1 Importance Of Domain Origin

An Automatic Consideration Of Domain Root Has A Good Impact On Strategies Not Only Involved In Revisiting Policy (.Edu And .Gov Domains Are Less Frequently Updated Than .Com [9]), But Also In Search Query Relevance Computation, Because Web-Site Geographical Location Matters If User Is From The Same Region, Language Or Culture.

That Is Why A Crawler May Want To Inspect Data, Such As

- WHOIS Information

- Page Original Language
- Server Geographical Location

For Example Hilltop Algorithm Ranks Links To External Domains With Higher Grades If They Are Geographically Closer To The Source. As Example, Domains From The Same University Network Have Higher Page Rank Than Pages From Mass-Media From The Same Region.

2.1.2 Evaluation Of Site Structure

Domain System Created For Ease Of Use By Humans And Tied With IP Resolving, Serves Now A More Commercial Role Than Before - A Three-Letter Domain Is A Valuable Resource And A Good Example Of How Popularity Affects Ranking.

Page Level In Domain Hierarchy Is The Number Of Links User Has To Click From The Root Page. Higher Level Pages Are Easier To Access. Pages, Which Cannot Be Accessed By Means Of Hyperlinks, Are *Robot-Isolated*. Pages Which Are Visually Not Accessible For Human Observer Are *Human-Isolated* Pages.

Considering Page Location As An Independent File And How Deep It Is Located In URL (Divided By / Symbol) Takes Place From Old Ways, When Html Files Were Static. This Has No Effect Now, Since Dynamic Server-Scripting Is In Effect And Web-Engines Try To Use One Gateway (Index.Php File For Example).

Ranking A Page Based On Its Level Comes From A Psychological Assumption That A Random User Is Most Likely To Type Short Domain Address Than Entire And Exact Path To The Required Document. By Doing So It Is Possible, That More Related Information Will Be Found In The Process.

Page Level Cannot Be Speculated Though – Like In Every Hierarchy, A Higher Level Element Is More Probable To Have Relevant Information The User Seeks, But At The Same Time, Due To Higher Abstraction Level, It Cannot Give More Specific Details. It Will Also Be Noted Further, That Pages With Higher Level Tend To Update More Frequently.

2.1.3 Calculation Of Page Weight

Page Weight Calculation Is One Of The Ways To Measure Its Popularity. Since Every Web-Site Fights For Visitors, Calculation Algorithm Must Be Smart Enough To Distinguish Documents Made For Engines From Those That Are Made For Visitors.

Page Rank, Hypertext-Induced Topic Selection , On-Line Page Importance Computation , Topic Citation Index Are Algorithms, Which Are Focused On Creating **Labeled Weighted Graph**, Where Each Vertex And Edge Is Associated With A Value Or Several Values, Like Width, Weight, Length. Vertex Weight Is Calculated Based On The Number Of Incoming And Outgoing Edges.

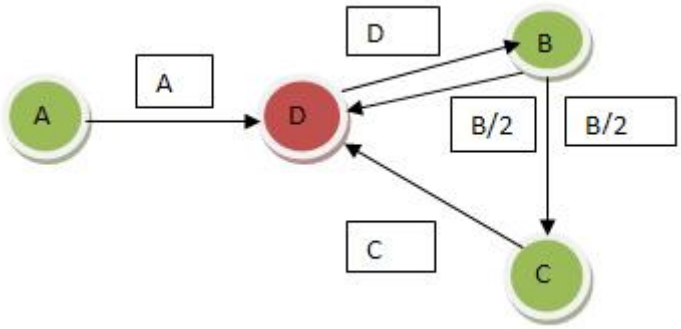

Figure 2. Page Rank Computation Example

Every Algorithm Has Its Differences; Further The Author Explains How Simplified Page Rank Is Done. Consider There Is A Small Graph (Figure 2) With Nodes Which Have Some Variable Ranks (Named After Vertex Names). Every Directed Edge This Way Would Give A Surplus Of Its Weight To A Vertex It Is Pointing To, Ergo Vertex B Has Some Page Rank B That Is Equally Divided Between All Links.

It Is Clear That If A Gives Entire Page Rank To D, Then A Would Have 0 Page Rank, Which Is Not Exactly A Right Thing, That's Why A Limiting Factor Was Introduced – *Damping Factor*, Which Basically States That Some Part Of Vertex Weight Stays With The Source. This Can Be Seen As Observer's Physical Limitation Of Viewing Links With Unlimited Deepness And Statistically Is Introduced As 15 %

The Calculation Of A,B,C,D Real Values Can Be Done Iteratively And Be Represented As A Matrix [16]. In The Calculation Process New Page Rank Does Not Depend On Its Previous Value.

The Main One-Iteration Simplified Formula For Calculating Page Rank For Vertex A Is

Iteration Process Is Shown On Figure 4.

R1= R2= R3= R4=

Figure 3. Page Rank Matrix Change With Each Iteration

Page That Has No Incoming Link Has Eventually Page Rank 0.15. New Pages That Get Indexed Automatically Gain Weight Of 1. With Each Iteration, The Page Rank Normalizes, Yet The Differences In Rank Values Of Different Pages Are On Logarithmic Scale.

A More Simple Way To Order Graph, Is To Consider Tree Structure Of The Web And To Use Sum Number Of Incoming Links To The Current Page From Different Domains. Major Disadvantage Is A Possibility That Someone Decides To Create A Link Farm On Several Domains.

2.2 Crawling Policy

The Behavior Of A Web Crawler Is The Outcome Of A Combination Of Policies:

- A *Selection Policy* That States Which Pages To Download,
- A *Re-Visit Policy* That States When To Check For Changes To The Pages,
- A *Politeness Policy* That States How To Avoid Overloading Web Sites, And
- A *Parallelization Policy* That States How To Coordinate Distributed Web Crawlers.

2.2.1 Selection Policy

This Involves The Selection Of Algorithms That Will Be Used To Enforce Selection Policy, It Requires A Metric Of Importance For Prioritizing Web Pages. The Importance Of A Page Is A Function Of Its Intrinsic Quality, Its Popularity In Terms Of Links Or Visits, And Even Of Its URL (The Latter Is The Case Of Vertical Search Engines Restricted To A Single Top-Level Domain, Or Search Engines Restricted To A Fixed Web Site). Designing A Good Selection Policy Has An Added Difficulty: It Must Work With Partial Information, As The Complete Set Of Web Pages Is Not Known During Crawling.

Assuming That Every Internet Page Has Not Only Subjective Value But Objective Informational Value (Facts, References Etc.) As Well, Every Crawler Need A Scheduling Algorithm, Which Would Give High Priority Queue On Indexing Most Valued Pages.

Crawler Is A Part Of Search-Engine Or Other Info System That Has To Request Information First, And Then Manage It. Similarity With Astronomy Is Very Close. The Observer First Has To Choose What Part Of The Sky He Wants To Observe, Then Questions Of Revisiting, Politeness And Parallelization Start To Form As Policies.

The Same It Is With The Internet. It Is Hard To Choose What Pages Should Be Travelled First, Hard To Distinguish How Important They Are, What Is Their Age And How Frequently They Change, How Often To Visit Them Without Raising Questions With Server Overloads.

Starting With A Seed, Crawler, If Not Being Limited, Will Eventually Find More And More Links. Depending On The Pages-Per-Domain Limit, This Ratio May Vary Around 3 New Links Per Page And Considering Virtually Unlimited (Over 100 Billion Pages) Internet Galaxy, It Is Critical To Use The Right Algorithms To Receive Right Information In Time.

2.2.1.1 First-In-First-Out Frontier (Aka Breadth-First)

Considered The Most Simple And Reliable, FIFO Algorithm Looks Through Documents That Were Added First, Creating A Queue Of Urls To Look Through. In Case Repository Is Full And No New Links Are Found, FIFO Algorithm Tells Crawler To Look Through Oldest Documents. This Makes Sense In General, Since Otherwise Some Kind Of Ordering, Planning And Analysis Is Required, Which Is Often Not Processing-Friendly.

Advanced FIFO Algorithms Include Increased Attention To Time-Modified Response In HTTP Request Header And To Weights Of The Graph Node. Simplified, This Makes Front Pages Of The Domain To Be Updated More Frequently Than Deeper Pages. Though Some Domains Can Be Considered Subjectively More Important And May Gain Certain Weight By Overriding General Rules. This Is Often Done In Mass-Media And Blog Services, Where Quick Partial Indexing Is More Important Than Full Sweep.

2.2.1.2 Context-Dependent Frontier

A Simple FIFO Crawling Algorithm Does Not Take Into Account Link Context, As Human Does. For Human Eye Text With Small Font Or Low Contrast Is Not Considered Important. On The Contrary, Blinking Images, Bold Text Or Text Headings Focus Observer's Attention. And Attention Means That This Part Of Document Has Higher Value, Higher Probability For Links To Be Clicked.

2.2.1.3 Weight-Focused Frontier

Page-Rank Based Selection And Uniform-Cost Search Algorithms Use Page Weight To Evaluate What Pages Have Priority To Be Crawled More Often. Some Weight Value Is Considered To Be Equal To Frequency Value. The Conversion Is Done By Evaluating Size Of The Entire Database, Sum Of All Weights And Estimation Of How Quick Crawling Is Done.

2.2.1.4 Harvesting Entire Domain

Since Domain Has A Tree-Structure, Its Source Can Be Acquired In Several Ways. Using These Algorithms In General Way Of Indexing Pages Without Paying Attention To Selected Domain Is Not A Smart Move, Since Some Of Them Focus On Specific Areas Of Crawling.

- **Breadth-First Search** (Figure 4) Focuses On Indexing Pages As They Appear On The First Levels Of The Page. Each Link Is Inserted In The Queue To Be Traversed Later. This Is The Most Simple And Obvious Way Of Differentiating Pages By Deepness, As User Has To Follow Several Pages To It. A Limitation Can Be Either Stopping At Certain Level, Or/And At Certain Number Of Pages Or May Implement **Beam-Search** Algorithm By Limiting Number Of Pages To Be Evaluated On Every Level.

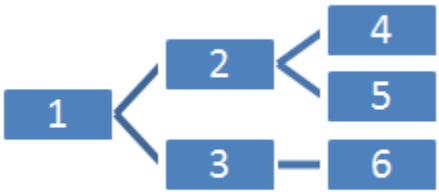

Figure 4. Breadth - first search example

- **Depth-First Search** (Figure 5) Relies On The Need Of Quick Indexing Of Deep Resources. Though It Does Suffer From The Problem Of Infinite Nodes, Often Produced By Auto-Generation Of Pages, It Has Its Advantages Of Simulating Human Behavior By Visiting Documents Which Lie Deeper, As Human Would Do If He Knew What He Looks For. A Solution Is To Declare Maximum Level Of Traversing.

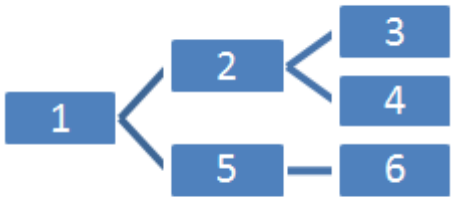

Figure 5. Depth-first search example

Reversed Uniform-Cost Search (Figure 6) Has Graph Weight Attribute Associated With Each Vertex. Root Vertex Is Given Maximum Weight, Which Is Equally Distributed Among Child Vertices Recursively. Vertices With Maximum Weight Are Crawled First, Since Algorithm Assumes That User Does Not Know What Information He Needs, Or Where It Is Located, Due To Misleading Structure Of The Web-Site, Because Of What High Accessibility Pages Are Viewed First. As Opposition To This Assumption, Crawling Through Minimum Weight Pages Assumes That User Knows Exactly What He Looks For And Where It Is Located.

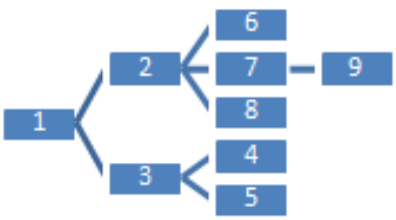

Figure 6. Cost-first search example

Best-First Search Assumes That Crawler Has Heuristics Function That Evaluates All Relevant Data And Decides Which Link To Index. This May Include Link Ordering By Its Location In The Document, Bayesian Statistics Calculation That Page Itself Is Not Used To Fraud Crawler, Using Global Page Weight (Back-Links), Even **Randomization** Of Links, Incorporation Of Other Algorithms Etc.

2.2.1.5 Focused Crawling

Major Disadvantage Of Algorithms Listed Above Is That Selection Policy Has Only One Purpose – To Cover As Much Documents As Possible, Even If They Are The Same. Focused Crawling Is Different – Its Selection Policy Depends On A Set Of Dictionary Keywords. The Page Which Keyword Subset Is Found To Coincide Statistically With The Dictionary Is Accepted. This Method Is Also Known As Bayesian Filtering And Is Used In Email Spam Distinction.

Some Focused Crawling Algorithms Update Dictionary With Found Top Keywords, Increasing Its Agility, While Others Use A Database Of Indexed Resources (Search Engines Like Google.Com) To Find Pages That Link To This Document And Assume That They Have Topic In Common. Event When Focused Crawling Is Not Required As Main Strategy, It May Improve General Crawling. Trust Rank Algorithm Focuses On Finding Pages And Links, Which Should Not Be Crawled.

Although Focused Crawling Has Good Perspectives, Since Relevance Is Very Important In Search Engine Life, It Has Its Drawbacks – Spam Filters Require Training.

2.2.2 Re-Visit Policy

The Web Has A Very Dynamic Nature, And Crawling A Fraction Of The Web Can Take Weeks Or Months. By The Time A Web Crawler Has Finished Its Crawl, Many Events Could Have Happened, Including Creations, Updates And Deletions.

Freshness: This Is A Binary Measure That Indicates Whether The Local Copy Is Accurate Or Not. The Freshness Of A Page *P* In The Repository At Time *T* Is Defined As:

$$F_p(t) = \begin{cases} 1 & \text{if } p \text{ is equal to the local copy at time } t \\ 0 & \text{otherwise} \end{cases}$$

Age: This Is A Measure That Indicates How Outdated The Local Copy Is. The Age Of A Page *P* In The Repository, At Time *T* Is Defined As:

$$A_p(t) = \begin{cases} 0 & \text{if } p \text{ is not modified at time } t \\ t - \text{modification time of } p & \text{otherwise} \end{cases}$$

The Objective Of The Crawler Is To Keep The Average Freshness Of Pages In Its Collection As High As Possible, Or To Keep The Average Age Of Pages As Low As Possible. These Objectives Are Not Equivalent: In The First Case, The Crawler Is Just Concerned With How Many Pages Are Out-Dated, While In The Second Case, The Crawler Is Concerned With How Old The Local Copies Of Pages Are.

Two Simple Re-Visiting Policies Were Studied By Cho And Garcia-Molina:

Uniform Policy: This Involves Re-Visiting All Pages In The Collection With The Same Frequency, Regardless Of Their Rates Of Change.

Proportional Policy: This Involves Re-Visiting More Often The Pages That Change More Frequently. The Visiting Frequency Is Directly Proportional To The (Estimated) Change Frequency.

(In Both Cases, The Repeated Crawling Order Of Pages Can Be Done Either In A Random Or A Fixed Order.)

Cho And Garcia-Molina Proved The Surprising Result That, In Terms Of Average Freshness, The Uniform Policy Outperforms The Proportional Policy In Both A Simulated Web And A Real Web Crawl. Intuitively, The Reasoning Is That, As Web Crawlers Have A Limit To How Many Pages They Can Crawl In A Given Time Frame, (1) They Will Allocate Too Many New Crawls To Rapidly Changing Pages At The Expense Of Less Frequently Updating Pages, And (2) The Freshness Of Rapidly Changing Pages Lasts For Shorter Period Than That Of Less Frequently Changing Pages. In Other Words, A Proportional Policy Allocates More Resources To Crawling Frequently Updating Pages, But Experiences Less Overall Freshness Time From Them.

2.2.3 Politeness Policy

Crawlers Can Retrieve Data Much Quicker And In Greater Depth Than Human Searchers, So They Can Have A Crippling Impact On The Performance Of A Site. Needless To Say, If A Single Crawler Is Performing Multiple Requests Per Second And/Or Downloading Large Files, A Server Would Have A Hard Time Keeping Up With Requests From Multiple Crawlers.

As Noted By Koster, The Use Of Web Crawlers Is Useful For A Number Of Tasks, But Comes With A Price For The General Community. The Costs Of Using Web Crawlers Include:

- Network Resources, As Crawlers Require Considerable Bandwidth And Operate With A High Degree Of Parallelism During A Long Period Of Time;
- Server Overload, Especially If The Frequency Of Accesses To A Given Server Is Too High;
- Poorly Written Crawlers, Which Can Crash Servers Or Routers, Or Which Download Pages They Cannot Handle; And
- Personal Crawlers That, If Deployed By Too Many Users, Can Disrupt Networks And Web Servers.

A Partial Solution To These Problems Is The Robots Exclusion Protocol, Also Known As The Robots.Txt Protocol That Is A Standard For Administrators To Indicate Which Parts Of Their Web Servers Should Not Be Accessed By Crawlers. This Standard Does Not Include A Suggestion For The Interval Of Visits To The Same Server, Even Though This Interval Is The Most Effective Way Of Avoiding Server Overload.

2.2.4 Parallelization Policy

Once Crawling And Database Is Run In Industrial Way, Parallelization Is Inevitable In Order To Overcome Timeouts From Fetching Pages That Are Far-Far Away Or Do Not Exist At All And In Order To Increase Processing Speed And Bandwidth.

Centralized Way Of Implementing Parallelization Is To Use A Front-End Server, Which Will Manage All Processes, Resolve DNS Names And Pass Them On To Processes That Await Response. Obvious Problem Occurs If Central Planning Server Is Down Or Is Unreachable; Otherwise It Is A Good Way Of Managing Database Without Creating Repeated Requests Of The Same URL From Multiple Processes. A Simple Example Is To Store A Number Of Threads And Assign Each Thread Its Number, Which Would Affect The URL To Be Fetched.

Decentralized Processes Manage Entire Cycle On Their Own, Though They Must Have Some Static Way Of Self-Management To Minimize Same URL Re-Fetching. In This Work The Author Uses Selection Policy With Randomization Algorithm, Which Makes Sure That Probability Of Requesting The Same Page By Two Or More Crawling Processes Is Minimized.

Parallelization Is Not Only Required To Increase Processing Speed, But Also Is A Good Way To Combine Different Strategies Of Search Engine. These Strategies May Include Resource Type Focus, Independent Language And Revisit Policies.

III. Discussion

According To Alex Faaborg Of The Mozilla Foundation, Future Browsers Will Likely Evolve Into Information Brokers, Helping Humans In Their Everyday Life. Semantic-Web Agent Will Have More Rich Access To Resources On The Net And On The Desktop. Joining Services With Single API Is Either An Idée Fix, Or A Panacea, Which Still Continue To Develop.

W3Consorcium States That For Knowledge Management, Web Masters Can Use RDF With OWL Support Which Can Describe Entities, Connections And Properties (Figure 7). But Its Development Along With SPARQL Is So Specific, That Most Of Modern Solutions Are Still Based On XML, RSS And SOAP, And Market Chooses Itself What Technology Is Better.

Microformats Is Another Solution To Exchanging Data Among Different Applications. These Constructions Are Embedded In HTML Source, Have Open API And May Be Readable By Human Observer As Well. Microformats Cover Different Topics – From Curriculum Vitae To Genealogy, And Are Developed By Independent Authors. The Disadvantage Is That Each Format Has Its Own Structure And Data Extraction Is Hard To Implement For All Of Them In One Application.

Another Speculation Is That Global Search Engines Will Likely Evolve To Artificial Intelligence, Based On Text Recognition At First, Reaching To Image, Sound And Video. A Web-Crawler In This Context Would Probably Be Used As A Sensor, Connecting Real-Life World With First Layers Of AI.

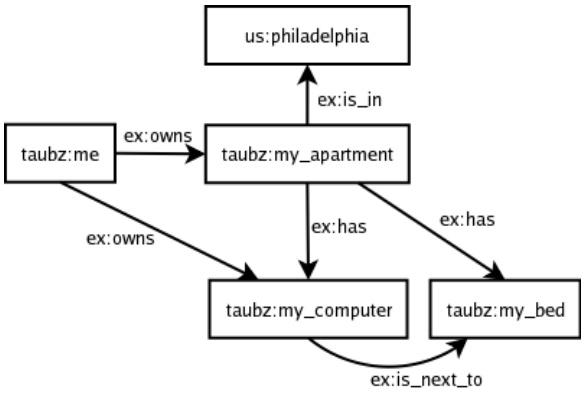

Figure 7. RDF As Graph. Source: Xmlhack.Ru

3.1 Features Of Web-Crawler

From The Study, It Is Obvious That A Web Crawler Must At A Minimum Possess Some Qualities, This Makes It Possible For Developers To Create Tailor Made Solutions For Specific Occasions. But In General, A Web Crawler Should Possess:

- Agility. A Crawler That Is Easy To Modify Is Crucial To Extend System Quickly, Without Losing Time On Modifying Entire Application. This Also Includes Ease Of Integration In Other Systems And Data Exchange With Applications That May Require Access To The Database For Visualization Or Other Needs.
- Robustness And Speed. Crawling Has To Withstand Anomalies That Occur With Certain Pages, Domains Or Connections And Must Be Optimized So That Cost-Per-Performance Would Be As Little As Possible. Speed Increase In The Future Should Be Done Using Distributed Processes. Once Limitations Are Specified, Seed Links Can Be Added And Crawler Should Start Its Cycle, Boosted By Multithreading System. The Number Of Simultaneous Threads Is Available For User To Change.
- Manageability. Configuration Is Not Written Solely In Code - It Is Dynamic And User May Change Limitations, Crawling Etiquette And Activity. Crawling Cycle Console Is Also Provided For User To Monitor Process Status. Crawler User Interface Should Be Simple Enough To Specify Limitations Of The Crawling Method; Either It Is Entire Domain, Part Of The Domain Or Domain With Some Parts Of External Domains.
- Structure Analysis. Web Crawler Not Only Indexes The Web, It Also Tries To Build A Domain Structure – How Deep Every Page Is Located, What Response The Server Sent, What Are The 404-Pages. This Analysis Should Be Interesting For The Web-Site Managers, Since They Often Cannot See The Big Picture.
- Data Extraction. Crawler User Interface Provides A Possibility To Add Filters. Every Filter Is Used In Crawling Cycle To Find Data, Matching Its Criteria. Data Is Then Saved And Is Possible To Save Independently Of The Source.
- Search System. Searching Among The Indexed Results Is Not As Complicated As A Search Engine System. This Option Is Only An Example Of Possible Extensions Of A Simple Crawler.

IV. Conclusion

The Main Objective Of This Appraisal Study Was To Throw Some Light On Web Crawling Algorithms That Are Used To Implement Selection Policies. We Have Also Discussed The Various Crawling Algorithms And The Researches Related To Respective Algorithms And Their Strengths And Weaknesses Associated. We Believe That The Algorithms Discussed In This Paper Are In Their Own Right Good And, Depending On The Particular Need Of The Surfer, Is Capable Of Handling Web Searches And Indexing. We Would Also Want To Add That No "One Size Fits All" Situation Exist In Terms Of Selecting The Best Search Algorithm.

References

[1] Pavalam S M, Jawahar M, Felix K Akorli, S V Kashmir Raja " Web Crawler In Mobile Systems" International Conference On Machine Learning (ICMLC 2011), Vol. , Pp

[2] Alessio Signorini, "A Survey Of Ranking Algorithms" Retrieved From Http://Www.Divms.Uiowa.Edu/~Asignori/Phd/Report/Asurvey- Of-Ranking-Algorithms.Pdf 29/9/2011

[3] Maurice De Kunder, "Size Of The World Wide Web", Retrieved From Http://Www.Worldwidewebsize.Com/8/8/11

[4] Ricardo Baeza-Yates, Ricardo Baeza-Yates "Crawling A Country: Better Strategies Than Breadth- First For Web Page Ordering" , Proc. WWW 2005.

[5] Marc Najork, "Web Crawler Architecture" Retrieved From

Http://Research.Microsoft.Com/Pubs/102936/Edswebcrawlerarchitecture. Pdf Accessed On 10/8/11
[6] Junghoo Cho And Hector Garcia-Molina "Effective Page Refresh Policies For Web Crawlers" ACM Transactions On Database Systems, 2003.
[7] Steven S. Skiena "The Algorithm Design Manual" Second Edition, Springer Verlag London Limited, 2008, Pg 162.
[8] Ben Coppin "Artificial Intelligence Illuminated" Jones And Barlett Publishers, 2004, Pg 77.
[9] Andy Yoo,Edmond Chow, Keith Henderson, William Mclendon, Bruce Hendrickson, Äumit Catalyäurek "A Scalable Distributed Parallel Breadth-First Search Algorithm On Bluegene/L" ACM 2005.
[10] Alexander Shen "Algorithms And Programming: Problems And Solutions" Second Edition Springer 2010, Pg 135
[11] Narasingh Deo "Graph Theory With Applications To Engineering And Computer Science" PHI, 2004 Pg 301
[12] Sergey Brin And Lawrence Page "Anatomy Of A Large Scale Hypertextual Web Search Engine" Proc. WWW Conference 2004
[13] Yongbin Qin And Daoyun Xu "A Balanced Rank Algorithm Based On Pagerank And Page Belief Recommendation"
[14] TIAN Chong "A Kind Of Algorithm For Page Ranking Based On Classified Tree In Search Engine" Proc International Conference On Computer Application And System Modeling (ICCASM 2010)
[15] J.Kleinberg "Authoritative Sources In A Hyperlinked Environment", Proc 9th ACM-SIAM Symposium On Discrete Algorithms, 1998.
[16] Shaojie Qiao, Tianrui Li, Hong Li And Yan Zhu, Jing Peng, Jiangtao Qiu "Simrank: A Page Rank Approach Based On Similarity Measure" 2010 IEEE
[17] S. N. Sivanandam, S. N. Deepa "Introduction To Genetic Algorithms" Springer, 2008, Pg 20
[18] S.N. Palod, Dr S.K.Shrivastav,Dr P.K.Purohit "Review Of Genetic Algorithm Based Face Recognition"
[19] Deep Malya Mukhopadhyay, Maricel O. Balitanas, Alisherov Farkhod A.,Seung-Hwan Jeon, And Debnath Bhattacharyya "Genetic Algorithm: A Tutorial Review" International Journal Of Of Grid And Distributed Computing Vol.2, No.3, September, 2009.
[20] Shian-Hua Lin, Jan-Ming Ho, Yueh-Ming Huang ,ACRID ,Intelligent Internet Document Organization And Retrieval ,IEEE Transactions On Knowledge And Data Engineering, 14(3),559-613, 2002

WEBSITES

[1] Whois.Net Http://Www.Whois.Net/
[2] Technorati Http://Technorati.Com/Weblog/2007/04/328.Html
[3] How Much Information? 2003 Http://Www2.Sims.Berkeley.Edu/Research/Projects/How-Much-Info-2003/Internet.Htm Internet Growth Statisticshttp://Www.Internetworldstats.Com/Emarketing.Htm
[4] How Google Works? Http://Www.Baselinemag.Com/Print_Article2/0,1217,A=182560,00.Asp
[5] Internet Archive Http://Www.Archive.Org/Index.Php
[6] Effective Web Crawling, Castillo C. 2004 [Online] Http://Www.Dcc.Uchile.Cl/~Ccastill/Crawling_Thesis/Effective_Web_Crawling.Pdf
[7] Hilltop Algorithm Http://Pagerank.Suchmaschinen-Doktor.De/Hilltop.Html Hilltop: A Search Engine Based On Expert Documents, Krishna Bharat; George A. Mihaila Http://Www.Cs.Toronto.Edu/~Georgem/Hilltop/
[8] The Anatomy Of A Large-Scale Hypertextual Web Search Engine; 1998 Brin, S.; Page, L. Http://Dbpubs.Stanford.Edu:8090/Pub/1998-8
[9] Method And System For Identifying Authoritative Information Resources In An Environment With Content-Based Links Between Information Resources. Kleinberg J.M. 2000 Http://Patft.Uspto.Gov/Netacgi/Nph-Parser?Sect1=PTO1&Sect2=HITOFF&D=PALL&P=1&U=%2Fnetahtml%2FPTO%2Fsrchnum.Htm&R=1&F=G&L=50&S1=6112202.PN.&OS=PN/6112202&RS=PN/6112202
[10] Adaptive On-Line Page Importance Computation. Abiteboul S., Preda M., Cobena G. 2003 Http://Www2003.Org/Cdrom/Papers/Refereed/P007/P7-Abiteboul.Html
[11] How Google Finds Your Needle In The Web's Haystack. Austin D. Grand Valley State University Http://Www.Ams.Org/Featurecolumn/Archive/Pagerank.Html
[12] ESP Game Http://Www.Espgame.Org/
[13] Focused Crawling: A New Approach To Topic-Specific Web Resource Discovery. S. Chakrabarti, M. Van Der Berg, And B. Dom. 1999.
[14] Focused Crawling Using Context Graphs. M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles And M. Gori, 2000 [Online] Http://Clgiles.Ist.Psu.Edu/Papers/VLDB-2000-Focused-Crawling.Pdf
[15] Combating Web Spam With Trustrank. Gy¨Ongyi Z., Garcia-Molina H., Pedersen J. Http://Www.Vldb.Org/Conf/2004/RS15P3.PDF
[16] MIME Types List Http://Www.Iana.Org/Assignments/Media-Types/
[17] Search Engines And Web Dynamics. Risvik K.M., Michelsen R., 2002 Http://Citeseer.Ist.Psu.Edu/Cache/Papers/Cs/26004/Http:Zszzszwww.Idi.Ntnu.Nozsz~Algkonzszgenereltzszse-Dynamicweb1.Pdf/Risvik02search.Pdf
[18] WEB Lancer Http://Weblancer.Net/Projects/
[19] Mozilla Does Microformats: Firefox 3 As Information Broker Http://Www.Readwriteweb.Com/Archives/Mozilla_Does_Microformats_Firefox3.Php
[20] W3C RDF Http://Www.W3.Org/RDF/
[21] W3C SPARQL Http://Www.W3.Org/TR/Rdf-Sparql-Query/
[22] Microformats [WWW] Http://Microformats.Org/
[23] Itemaps Standart [WWW] Http://Www.Sitemaps.Org/ (13.06.2007)
[24] Mercator: A Scalable, Extensible Web Crawler Heydon, A. And Najork, M. 1999 Http://Www.Cindoc.Csic.Es/Cybermetrics/Pdf/68.Pdf
[25] Search Engines And Web Dynamics.Risvik, K. M. And Michelsen, R. 2002. Http://Citeseer.Ist.Psu.Edu/Rd/1549722%2C509701%2C1%2C0.25%2cdownload/Http://Citeseer.Ist.Psu.Edu/Cache/Papers/Cs/26004/Http:Zszzszwww.Idi.Ntnu.Nozsz%7ealgkonzszgenereltzszse-Dynamicweb1.Pdf/Risvik02search.Pdf

Oni O. A. "An Appraisal of Web Crawling Algorithms"International Journal of Engineering Science Invention (IJESI), vol. 07, no. 04, 2018, pp 42-50