

# **SEARCH-ENGINE : SPORTS(CRICKET)**

## **Crawling : Harsh Verma**

### **TECH STACK USED:**

1. Apache Nutch-1.15
2. Solr-7.3.1d
3. Selenium-2.42.2
4. Firefox-29.0
5. Java 8.0

**Apache Nutch is an Internet search engine software, web Crawler, powerful for vertical search engine :**

For crawling web pages associated with Cricket Sports, The Apache Nutch Framework was utilized for crawling as well as feeding the Fetched Content from crawling to the Solr Framework hosted on localhost for indexing the fetched web pages as well as creating web graphs for implementing Page Rank and the HITS algorithms.

### **Why Use Nutch ?**

- Production ready Web Crawler, Scalable , Tried and Tested
- Fine grained Configurations
- Relying on Apache hadoop data structure, batch processing
- MultiThreaded.
- Allows Custom Implementation for parse, index and scoring.
- Pluggable Indexing (solr, mongodb, elastic search, etc)
- Automation on checking broken links
- Handle Duplication
- User friendly ,Url Filters, Normalization

### **Crawled Data Numbers:**

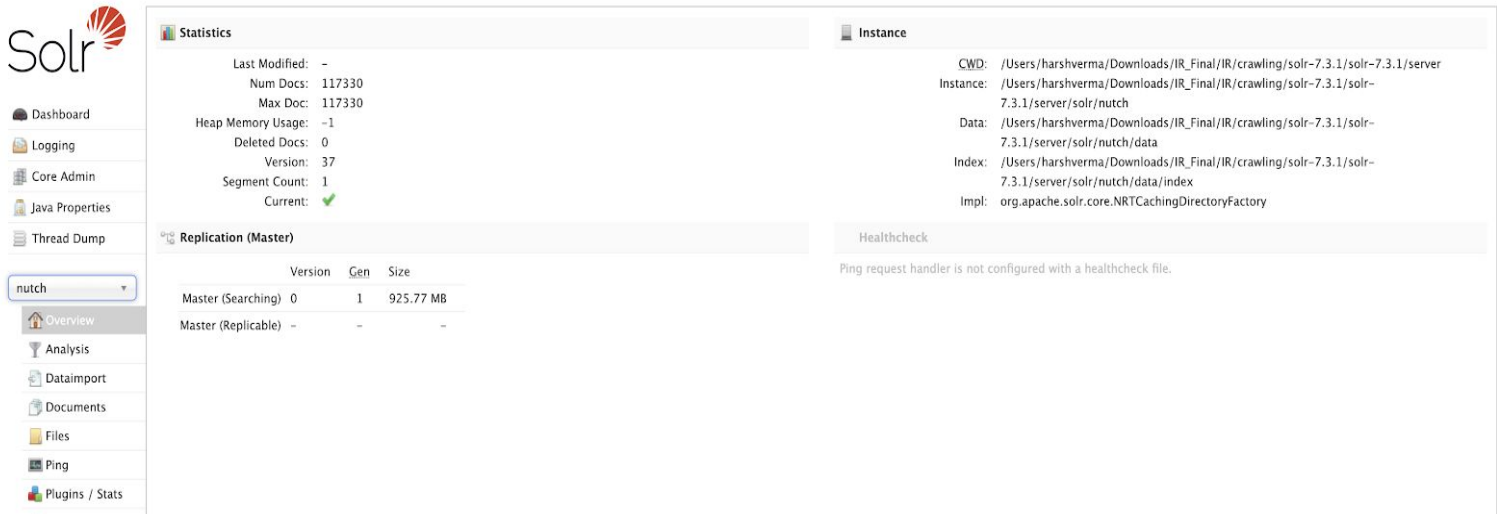
**The number of pages crawled are : 1,17,330**

**The number of web pages crawled & fetched are : 1,20,120**

Name: HARSH VERMA

Net ID: hxv180001

## Screenshot of Documents Stats in solr :



The screenshot displays the Solr Admin interface for the 'nutch' core. The left sidebar contains navigation links: Dashboard, Logging, Core Admin, Java Properties, Thread Dump, and a dropdown menu with Overview, Analysis, Dataimport, Documents, Files, Ping, and Plugins / Stats. The main content area is divided into three sections: Statistics, Replication (Master), and Instance.

**Statistics**

Last Modified:	-
Num Docs:	117330
Max Doc:	117330
Heap Memory Usage:	-1
Deleted Docs:	0
Version:	37
Segment Count:	1
Current:	✓

**Replication (Master)**

	Version	Gen	Size
Master (Searching)	0	1	925.77 MB
Master (Replicable)	-	-	-

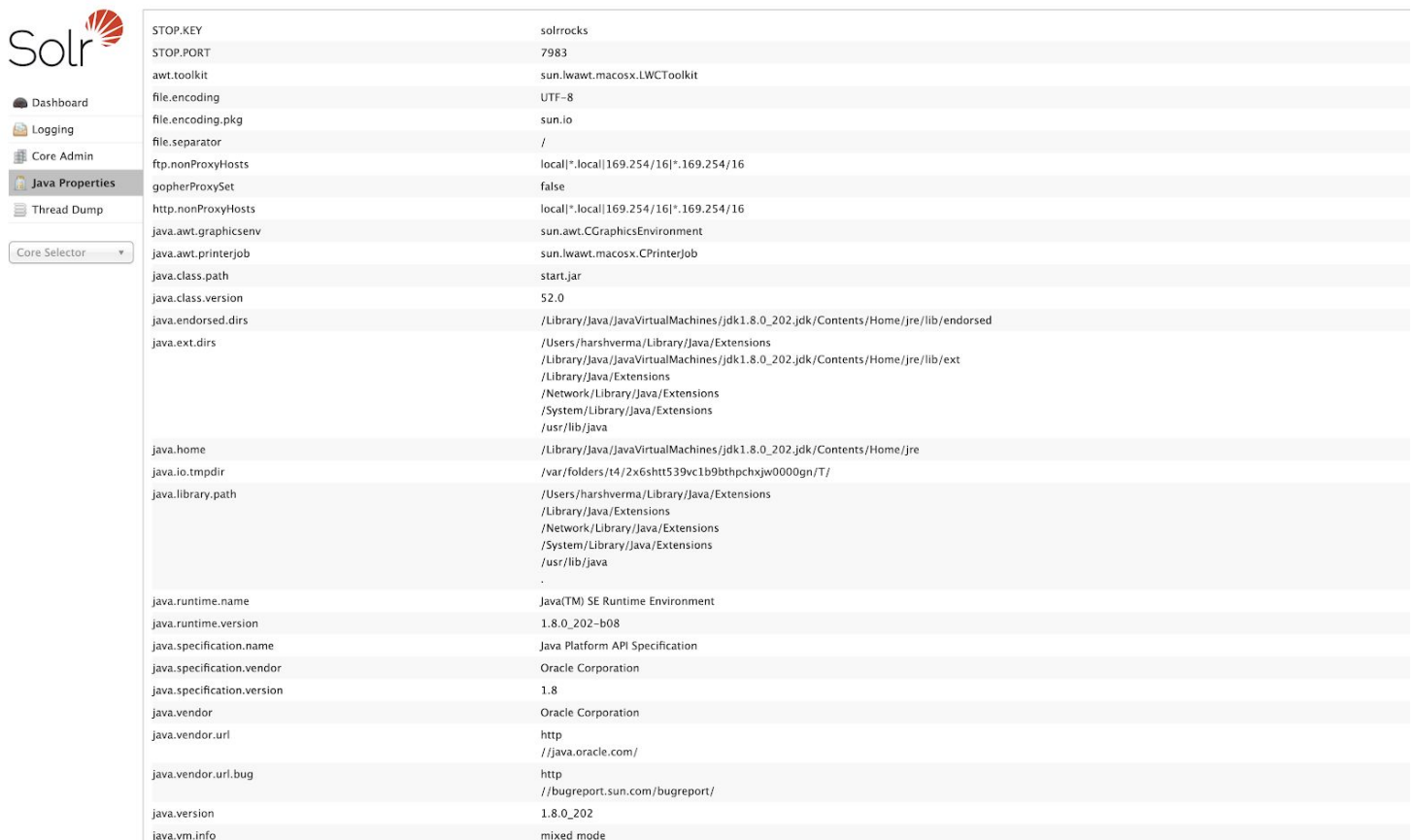
**Instance**

CWD: /Users/harshverma/Downloads/IR\_Final/IR/crawling/solr-7.3.1/solr-7.3.1/server  
Instance: /Users/harshverma/Downloads/IR\_Final/IR/crawling/solr-7.3.1/solr-7.3.1/server/solr/nutch  
Data: /Users/harshverma/Downloads/IR\_Final/IR/crawling/solr-7.3.1/solr-7.3.1/server/solr/nutch/data  
Index: /Users/harshverma/Downloads/IR\_Final/IR/crawling/solr-7.3.1/solr-7.3.1/server/solr/nutch/data/index  
Impl: org.apache.solr.core.NRTCachingDirectoryFactory

**Healthcheck**

Ping request handler is not configured with a healthcheck file.

## Solr Settings Screenshot :



The screenshot displays the Solr Admin interface for the 'Java Properties' settings. The left sidebar contains navigation links: Dashboard, Logging, Core Admin, Java Properties, Thread Dump, and a dropdown menu with Core Selector. The main content area shows a list of system properties and their values.

Property	Value
STOP.KEY	solrlocks
STOP.PORT	7983
awt.toolkit	sun.lwawt.macosx.LWCToolkit
file.encoding	UTF-8
file.encoding.pkg	sun.io
file.separator	/
ftp.nonProxyHosts	local[*].local 169.254/16 *.169.254/16
gopherProxySet	false
http.nonProxyHosts	local[*].local 169.254/16 *.169.254/16
java.awt.graphicsenv	sun.awt.CGraphicsEnvironment
java.awt.printerjob	sun.lwawt.macosx.CPrinterJob
java.class.path	start.jar
java.class.version	52.0
java.endorsed.dirs	/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/jre/lib/endorsed
java.ext.dirs	/Users/harshverma/Library/Java/Extensions /Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/jre/lib/ext /Library/Java/Extensions /Network/Library/Java/Extensions /System/Library/Java/Extensions /usr/lib/java
java.home	/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/jre
java.io.tmpdir	/var/folders/t4/2x6shtt539vc1b9bthpchxjw0000gn/T/
java.library.path	/Users/harshverma/Library/Java/Extensions /Library/Java/Extensions /Network/Library/Java/Extensions /System/Library/Java/Extensions /usr/lib/java .
java.runtime.name	Java(TM) SE Runtime Environment
java.runtime.version	1.8.0_202-b08
java.specification.name	Java Platform API Specification
java.specification.vendor	Oracle Corporation
java.specification.version	1.8
java.vendor	Oracle Corporation
java.vendor.url	http://java.oracle.com/
java.vendor.url.bug	http://bugreport.sun.com/bugreport/
java.version	1.8.0_202
java.vm.info	mixed mode

Searching Document in Solr:

Request-Handler (qt)

/select

common

q

\*:\*

fq

sort

start, rows

0

10

fl

df

Raw Query Parameters

key1=val1&key2=val2

wt

-----

☐ indent off

☐ debugQuery

☐ dismax

☐ edismax

☐ hl

☐ facet

☐ spatial

☐ spellcheck

Execute Query

http://localhost:8983/solr/nutch/select?q=:\*

```
{
  "responseHeader":{
    "status":0,
    "QTime":1,
    "params":{
      "q":":*:*",
      "_":":1556485916074"}},
  "response":{"numFound":51364,"start":0,"docs":[
    {
      "tstamp":"2019-04-23T04:39:12.134Z",
      "digest":"25b28a3e131366e8b5639d54756d4be1",
      "boost":0.82499284,
      "id":"http://www.cricwaves.com/mobile/loadArtD?aid=S9bIBk7piF_drh-sevawcirc&at=indian-t20-cricket-2019-desperate-rajasthan-face-confident-mumbai",
      "url":"http://www.cricwaves.com/mobile/loadArtD?aid=S9bIBk7piF_drh-sevawcirc&at=indian-t20-cricket-2019-desperate-rajasthan-face-confident-mumbai",
      "content":"Menu\\nSelect tour Indian T20 Cricket 2019 West Indies and Bangladesh in Ireland Tri-Series 2019 Pakistan tour of England 2019 Cricket Wo",
      "_version_":1631653672715812865},
    {
      "tstamp":"2019-04-23T01:24:16.011Z",
      "digest":"87e8020870fdd873912fe9c58d5284f0",
      "boost":0.82499284,
      "id":"http://www.cricwaves.com/mobile/loadArtD?aid=cfgJHtuX57_drh-sevawcirc&at=indian-t20-cricket-2019-can-delhi-shrug-off-kotla-woes-against-kings",
      "url":"http://www.cricwaves.com/mobile/loadArtD?aid=cfgJHtuX57_drh-sevawcirc&at=indian-t20-cricket-2019-can-delhi-shrug-off-kotla-woes-against-king",
      "content":"Menu\\nSelect tour Indian T20 Cricket 2019 West Indies and Bangladesh in Ireland Tri-Series 2019 Pakistan tour of England 2019 Cricket Wo",
      "_version_":1631653672716861441},
    {
      "tstamp":"2019-04-23T01:24:02.309Z",
      "digest":"f92842bdf09938552b513143bd66863e",
      "boost":0.82710844,
      "id":"http://www.cricwaves.com/mobile/loadArtD?aid=hdiMaNRQr8_drh-sevawcirc&at=captains-day-out-as-iyer-smith-shine",
      "url":"http://www.cricwaves.com/mobile/loadArtD?aid=hdiMaNRQr8_drh-sevawcirc&at=captains-day-out-as-iyer-smith-shine",
      "content":"Menu\\nSelect tour Indian T20 Cricket 2019 West Indies and Bangladesh in Ireland Tri-Series 2019 Pakistan tour of England 2019 Cricket Wo",
      "_version_":1631653672717910018},
    {
      "tstamp":"2019-04-23T03:03:53.052Z",
      "digest":"961eaa78f22a71571ca76bb05d241a80",
      "boost":0.40208387,
      "id":"http://www.cricwaves.com/mobile/loadCom?mid=4171&mt=australia-vs-pakistan-",
      "url":"http://www.cricwaves.com/mobile/loadCom?mid=4171&mt=australia-vs-pakistan-",
      "_version_":1631653672721055745,
      "content":"Menu\\nSelect tour Indian T20 Cricket 2019 West Indies and Bangladesh in Ireland Tri-Series 2019 Pakistan tour of England 2019 Cricket Wo",
    }
  ]}
}
```

**Apache Nutch generates 3 folders during the crawling operation:**

1. **CRAWLDB:** it maintains the information about URLs such as the fetch status, fetching schedule, metadata, etc.
2. **LINKDB:** For each URL, the LINKDB maintains the incoming and outgoing URLs for that URL which are further used to facilitate PAGE RANKING algorithm and the HITS algorithm.
3. **SEGMENTS:** contains multiple subdirectories within it. During Crawling, the crawl script creates multiple directory to store information for Crawl Fetching, Crawl Content, Crawl Parsing, Parsed Data and Parsed Text

**The Crawling Method can be described by the following methods:**

**STEP 1: INJECTOR**

- 1) The injector takes all the URLs of the seeds.txt file and adds them to the crawldb.
- 2) As a central part of Nutch, the crawldb maintains information on all known URLs (fetch schedule, fetch status, metadata).

**\$NUTCH\_RUNTIME\_HOME/bin/nutch inject crawl/crawldb urls**

**STEP 2: GENERATOR**

Based on the data of crawldb, the generator creates a fetchlist and places it in a newly created segment directory

**\$NUTCH\_RUNTIME\_HOME/bin/nutch generate crawl/crawldb crawl/segments -topN 130**

**STEP 3: FECTHER**

The fetcher gets the content of the URLs on the fetchlist and writes it back to the segment directory.

This step usually is the most time-consuming one , This respects the robots rules (robort.txt and robots directives)

**\$NUTCH\_RUNTIME\_HOME/bin/nutch fetch \$s1**

**STEP 4: PARSER**

Now the parser processes the content of each web page and for example omits all html tags. If the crawl functions as an update or an extension to an already existing one (e.g. depth of 3), the updater would add the new data to the crawlddb as a next step.

```
$NUTCH_RUNTIME_HOME/bin/nutch parse $s1
```

**STEP 5: LINK INVERTER**

Before indexing, all the links need to be inverted, which takes into account that not the number of outgoing links of a web page is of interest, but rather the number of inbound links. This is quite similar to how Google PageRank works and is important for the scoring function. The inverted links are saved in the linkdb.

```
$NUTCH_RUNTIME_HOME/bin/nutch invertlinks crawl/linkdb -dir crawl/segments
```

**STEP 6 and 7: SOLR INDEXER**

Using data from all possible sources (crawlddb, linkdb and segments), the indexer creates an index and saves it within the Solr directory.

For indexing, the popular Lucene library is used.

Now, the user can search for information regarding the crawled web pages via Solr.

```
bin/nutch index crawl/crawlddb/ -linkdb crawl/linkdb/ crawl/segments/*/ -filter -normalize  
-deleteGone
```

**The command used for creating the Dump is:**

```
bin/nutch readlinkdb Crawling/LinkDB -dump LinkDBDUMP
```

This generates a Dump containing the incoming and outgoing links for each URL.

**APACHE NUTCH CONFIGURATION**

**Modifications that were made to the nutch configuration file**

**Path:**apache-nutch-1.1.15/conf/nutch-site.xml

**Plugins:** protocol-http|protocol-httpclient|urlfilter-regex|index-(basic|more)|query-(basic|site|url|lang)|indexer-solr|nutch-extensionpoints|protocol-httpclient|urlfilter-regex|parse-(text|html|msexcel|msword|mspowerpoint|pdf)|summary-basic|scoring-opic|urlnormalizer-(pass|rege

x|basic)protocol-http|urlfilter-regex|parse-(html|tika|metatags)|index-(basic|anchor|more|metadata)  
)

**Fetcher Server Delay** (The number of seconds the fetcher will delay between successive requests to the same server): 1

Selenium Driver: firefox

**Http Redirect Max** The maximum number of redirects the fetcher will follow when trying to fetch a page. If set to negative or 0, fetcher won't immediately follow redirected URLs, instead it will record them for later fetching: 1

**Ignore outlinks to the same hostname:** False

Ignore outlinks to the same domain: False

Limit to only a single outlink to the same page: False

**This number is the maximum number of threads that should be allowed to access a queue at one time:** As per machine we set it to 40

### Screenshot of Nutch Config Properties: HTTP Agent and Plugin Properties

```
<configuration>
<property>
  <name>http.agent.name</name>
  <value>nutch-1.15-crawler</value>
</property>
<property>
  <name>storage.data.store.class</name>
  <value>org.apache.gora.mongodb.store.MongoStore</value>
  <description>Default class for storing data</description>
</property>
<property>
  <name>http.tls.certificates.check</name>
  <value>>false</value>
  <description>
    Whether to check the TLS/SSL server certificates for validity.
    If true invalid (e.g., self-signed or expired) certificates are
    rejected and the https connection is failed. If false insecure
    TLS/SSL connections are allowed. Note that this property is
    currently not supported by all http/https protocol plugins.
  </description>
</property>
<property>
  <name>plugin.includes</name>
  <value>protocol-http|protocol-httpclient|urlfilter-regex|index-(basic|more)
</property>
<property>
  <name>db.ignore.external.links</name>
  <value>>true</value>
</property>
<property>
  <name>db.ignore.external.links.mode</name>
  <value>byDomain</value>
</property>
```

**FTP Properties and Http Content Settings:**

```
<property>
  <name>ftp.server.timeout</name>
  <value>10000</value>
  <description>An estimation of ftp server idle time, in millisec.
Typically it is 120000 millisec for many ftp servers out there.
Better be conservative here. Together with ftp.timeout, it is used to
decide if we need to delete (annihilate) current ftp.client instance and
force to start another ftp.client instance anew. This is necessary because
a fetcher thread may not be able to obtain next request from queue in time
(due to idleness) before our ftp client times out or remote server
disconnects. Used only when ftp.keep.connection is true (please see below).
</description>
</property>
<property>
  <name>ftp.keep.connection</name>
  <value>false</value>
  <description>Whether to keep ftp connection. Useful if crawling same host
again and again. When set to true, it avoids connection, login and dir list
parser setup for subsequent urls. If it is set to true, however, you must
make sure (roughly):
(1) ftp.timeout is less than ftp.server.timeout
(2) ftp.timeout is larger than (fetcher.threads.fetch * fetcher.server.delay)
Otherwise there will be too many "delete client because idled too long"
messages in thread logs.
</description>
</property>
<property>
  <name>http.content.limit</name>
  <value>65536</value>
  <description>The length limit for downloaded content, in bytes.
If this value is nonnegative ( $\geq 0$ ), content longer than it will be truncated;
otherwise, no truncation at all.
</description>
</property>
<property>
  <name>generate.max.count</name>
  <value>-1</value>
</property>
</configuration>
```



**Firefox Driver and Dynamic Content Property**

```
<name>selenium.driver</name>
<value>firefox</value>
<description>
    A String value representing the flavour of Selenium
    WebDriver() to use. Currently the following options
    exist - 'firefox', 'chrome', 'safari', 'opera' and 'remote'.
    If 'remote' is used it is essential to also set correct properties for
    'selenium.hub.port', 'selenium.hub.path', 'selenium.hub.host' and
    'selenium.hub.protocol'.
</description>
</property>
<property>
    <name>selenium.firefox.headless</name>
    <value>true</value>
    <description>A Boolean value representing if firefox should
        run headless . make sure that firefox version is 55 or later,
        and selenium webDriver version is 3.6.0 or later. The default value is false.
        Currently this option exist for - 'firefox'
    </description>
</property>
<property> Add the plugin folders to your installation's NUTCH_HOME/src/plugin directory
    <name>selenium.take.screenshot</name>
    <value>false</value>
    <description>
        Boolean property determining whether the protocol-selenium
        WebDriver should capture a screenshot of the URL. If set to
        true remember to define the 'selenium.screenshot.location'
        property as this determines the location screenshots should be
        persisted to on HDFS. If that property is not set, screenshots
        are simply discarded.
    </description>
</property>
<property>
    <name>selenium.screenshot.location</name>
    <value></value>
    <description>
        The location on disk where a URL screenshot should be saved
        to if the 'selenium.take.screenshot' proerty is set to true.
        By default this is null, in this case screenshots held in memory
        are simply discarded.
    </description>
</property>
```



**Http Timeout and Robot Property Settings:**

```
<property>
  <name>http.timeout</name>
  <value>10000</value>
  <description>The default network timeout, in milliseconds.</description>
</property>
<property>
  <name>fetcher.threads.per.queue</name>
  <value>50</value>
  <description>This number is the maximum number of threads that
    should be allowed to access a queue at one time. Replaces
    deprecated parameter 'fetcher.threads.per.host'.
  </description>
</property>
<property>
  <name>file.content.ignored</name>
  <value>true</value>
  <description>If true, no file content will be saved during fetch.
    And it is probably what we want to set most of time, since file:// URLs
    are meant to be local and we can always use them directly at parsing
    and indexing stages. Otherwise file contents will be saved.
    !! NO IMPLEMENTED YET !!
  </description>
</property>
<property>
  <name>http.robots.403.allow</name>
  <value>true</value>
  <description>Some servers return HTTP status 403 (Forbidden) if
    /robots.txt doesn't exist. This should probably mean that we are
    allowed to crawl the site nonetheless. If this is set to false,
    then such sites will be treated as forbidden.
  </description>
</property>
<property>
  <name>http.max.delays</name>
  <value>10</value>
  <description>The number of times a thread will delay when trying to
    fetch a page. Each time it finds that a host is busy, it will wait
    fetcher.server.delay. After http.max.delays attempts, it will give
    up on the page for now.
  </description>
</property>
```

**STATISTICS FOR CRAWLED DATA:**

Nutch provides an api called readdb with stats as argument, which gives the stats of the data crawled :

TOPIC	NUMBER
Crawled Links	1,17,330
Db_unfetched	2,10,120 (Depth 2)
Db_gone	4320

**Common Issues Faced and Resolved:****1. Nutch do not fetch AJAX/JavaScript driven dynamic HTML content :**

-> In order to crawl webpages that rely on JavaScript/AJAX to dynamically load content we used the Selenium.

-> **The average rate Selenium used to crawl the pages is 500 pages / hour**

-> **So to crawl more than 1,00,000 pages it took around 200+ Hours, so we used multiple machines to achieve it in 10 days parallely.**

**2. Choosing Specific Version of Apache Nutch :**

I started with apache nutch 2.X , it has web-api where in we can schedule jobs and monitor. But 2.X is not stable and does not have as many features as 1.X like webgraph api and many more. Hence, I switched to nutch 1.5

**3. Filters :** Pages like blogs dummy pages, help pages where being crawled which are not needed, and where not crawled future with regex url filter.

**4. Selenium with Firefox compatibility issue:**

Selenium is automates browser. Nutch loads its pages in selenium's browser. So, It's very important to find the compatible version of browser with particular version of selenium. Version mentioned in requirements are compatible versions with nutch 1.5.

**5. Selenium has an threading issue of memory leak:** So due to this, many times browser gets stuck on a pages and do not terminate. Throwing port locked exception. Hence I run a command in cron for every 15-30 mins to kill firefox. This releases the locked ports, to be used by active threads.

## **SEEDS LIST**

Initially we gave 95 URLs as seed URL to start our crawling

**The sample list of seed URLs are as follows:**

<http://www.iplt20.com/>  
<http://www.icc-cricket.com/>  
<http://www.cricbuzz.com/>  
<http://www.cricket.com.au/>  
<http://www.royalchallengers.com/>  
<http://www.bcci.tv/>  
<http://www.cricketworld.com/>  
<http://chennaisuperkings.com/>  
<http://kkr.in/>  
<http://www.cricketnmore.com/>  
<http://www.mumbaiindians.com/>  
<http://cricwaves.com/>  
<http://www.pcb.com.pk/>  
<http://ecb.co.uk/>  
<http://www.kxip.in/>  
<http://cricketweb.net/>  
<http://rajasthanroyals.com/>  
<http://www.srilankacricket.lk/>  
<http://www.lords.org/>  
<http://www.cricket365.com/>  
<http://www.islandcricket.lk/>  
<http://cricket.af/>  
<http://yorkshireccc.com/>  
<http://www.lastmanstands.com/>  
<http://cricket.co.za/>  
<http://www.indiancricketfans.com/>  
<http://www.kiaoval.com/>  
<http://cricschedule.com/>  
<http://www.windiescricket.com/>  
<http://www.cricketvictoria.com.au/>  
<http://mumbaicricket.com/mca/>  
<http://www.cricbay.com/>  
<http://www.glamorgancricket.com/>

<http://emiratescricket.com/>  
<http://cricketcrowd.com/>  
<http://www.kentcricket.co.uk/>  
<http://zimcricket.org/>  
<http://pcboard.com.pk/>  
<http://hycricket.org/>  
<http://howstat.com.au/cricket/home.asp>  
<http://waca.com.au/>  
<http://cricschedule.com/>  
<http://cricruns.com/>  
<http://www.cricket.co.uk/>  
<http://www.worldcricketcentre.com/>  
<http://www.20-20.in/>  
<http://www.indiatimes.com/sports/>  
<http://www.in.com/sports/cricket/>  
<http://www.hotstar.com/sports/cricket>  
<http://sports.ndtv.com/cricket>  
<http://crickettimes.com/series/2312/IPL-2019/>  
<http://www.cricwaves.com/cricket/news/articles/>  
<http://www.onlinecricketbetting.net/blog/>  
<http://www.cricadium.com/category/ipl/2019-ipl-12/>  
<http://www.cricadium.com/category/cricket-world-cup/world-cup-2019/>  
<http://www.thefulltoss.com/>  
<http://caribbeancricket.com/news>  
<http://www.cricindeed.com/>  
<http://www.cricketnews.net.in/>  
<http://1tip1hand.com/>  
<http://www.thecricketblog.com/>  
<http://www.cricfirst.com/>  
<http://www.cricnmore.com/>  
<http://en.wikipedia.org/wiki/Cricket>  
[http://en.wikipedia.org/wiki/Indian\\_Premier\\_League](http://en.wikipedia.org/wiki/Indian_Premier_League)  
<http://cricket.yahoo.net/>  
<http://sportswiki.com/cricket>  
<http://betting.betfair.com/cricket/>  
<http://www.edailysports.com/category/cricket/>  
<http://www.sportseon.com/cricket/>  
<http://www.thecricketblog.info/>

## **HANDLING DUPLICATE DATA**

Apache Nutch handles duplication of the crawled data via its properties setting and commands

By default Nutch use the **org.apache.nutch.crawl.MD5Signature** class to calculate the digest of an URL, this class calculates the digest using the MD5Hash function of the raw binary content of the page, if no content is found then the URL is used.

The **DeduplicationJob** first groups fetched URLs by the digest (in your case both URLs should have the same signature/digest) and marks all the URLs as duplicated, except the one with the highest score, if both (or more) URLs have the same digest and the same score, then the one with the latest timestamp is used instead

## **HYPERLINK INFORMATION FOR INDEXING AND RELEVANCE MODEL**

Crawlddb of apache nutch has all the data that has been crawled. A small python script was run all the segments , and that script executed readseg api to get data from all the segments.

This api when called with parameters of to get title, url and content , created a dump for each segment.

A Python script was ran on this data to create, individual record files with Recno, Title, Url, Outlinks and Page Content which was then shared with Indexing person(Vatsal).

The content gathered by the python script from the data present in the segments has all the information such as:

- Record Number
- Title
- Urls
- Outlinks
- Page Content

**Python file Path:** apache-nutch-1.15/crawl\_parse/src/parse\_crawl.py

This output is then passed further for indexing and generating relevance model

## **SOLR :**

Every version of Nutch is built against a specific Solr version , **So the compatible version with Nutch-1.15 is Solr-7.3.1 .** We used Solr to monitor the data.

To start Solr

**`./bin/solr start`**

**Admin Console:** <http://localhost:8983/solr/#/>

In addition, filters, normalizers and plugins allow Nutch to be highly modular, flexible and very customizable throughout the whole process.

## **SELENIUM**

Nutch do not fetch AJAX/JavaScript driven dynamic HTML content

In order to crawl webpages that rely on JavaScript/AJAX to dynamically load content we used the Protocol-Selenium Plugin.

This plugin will load the pages that you're crawling in Selenium so that JavaScript will be handled properly.

So we used selenium -2.42.2 with Nutch which is the compatible version

With selenium -2.42.2 we used Firefox-29.

## **DISCUSSION**

We started implementing the crawler with Scrapy and BeautifulSoup. But, the issue that we faced was we were not able to get dynamic content from the websites as most of the websites uses JavaScript / Ajax to load the content. We decided to use open source crawler Apache-Nutch which was able to get dynamic content using a selenium plugin and firefox.

## **Monitoring for Nutch:**

Apache Nutch 1.X doesn't have web-ui , to monitor the jobs and logs error and manage crawling.

Hence, I created a email script where after every couple of crawl jobs, it reads the logs and send email to me, giving details of the jobs like :

- Number of jobs ran



- Number of urls processed.
- Amount of time each job took.
- Urls where Error Occurred , along with error
- Report failed Jobs.

**Screenshot of Email ->>>**

**Alter on CrawlJob : URGENT Error occurred**



**harshverma59@gmail.com**

to me ▼

Existing since previous crawl job is still in process

Indexer: 108 indexed (add/update)

Indexer: 112 indexed (add/update)

Indexer: 141 indexed (add/update)

Tue Apr 23 02:35:52 CDT 2019 : Iteration 1 of 1

Tue Apr 23 02:48:31 CDT 2019 : Iteration 1 of 1

Tue Apr 23 03:01:19 CDT 2019 : Iteration 1 of 1

## **CONCLUSION**

Apache Nutch is highly scalable, robust and relatively feature rich crawler. Quality – crawling can be biased to fetch "important" pages first. If we use Apache Nutch on AWS, Google Cloud Computing it can be clustered among 100's of machine at a time which can result in to fast performance or can be run in distributed mode using big-data mapreduce format.