Web Crawling

INF 141: Information Retrieval
Discussion Session
Week 4 – Winter 2008

Yasser Ganjisaffar

yganjisa@ics.uci.edu

Open Source Web Crawlers

Heritrix

Nutch

WebSphinx

Crawler4j

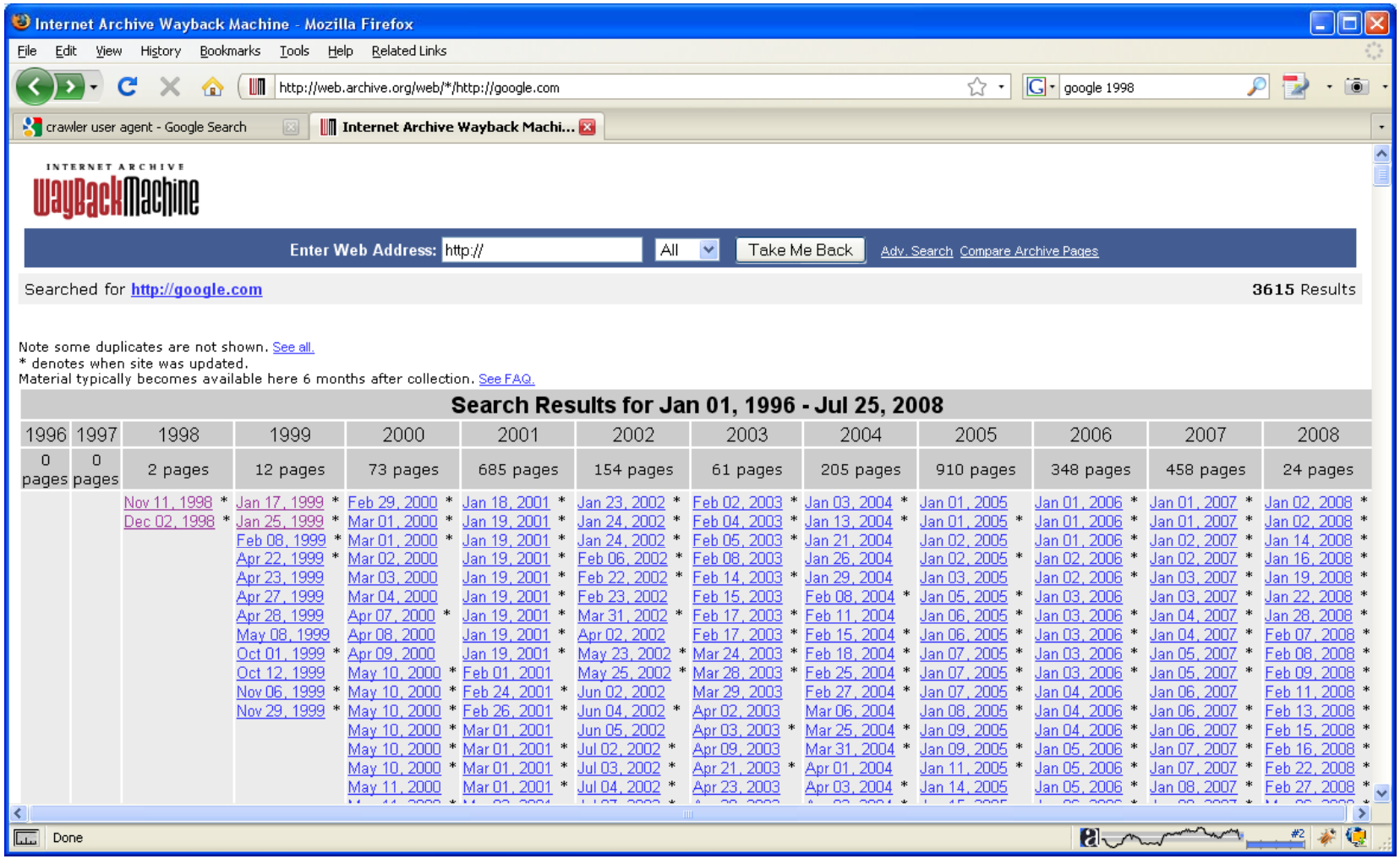
Heritrix

- Extensible, Web-Scale
- Command line tool
- Web-based Management Interface
- Distributed
- Internet Archive’s Crawler

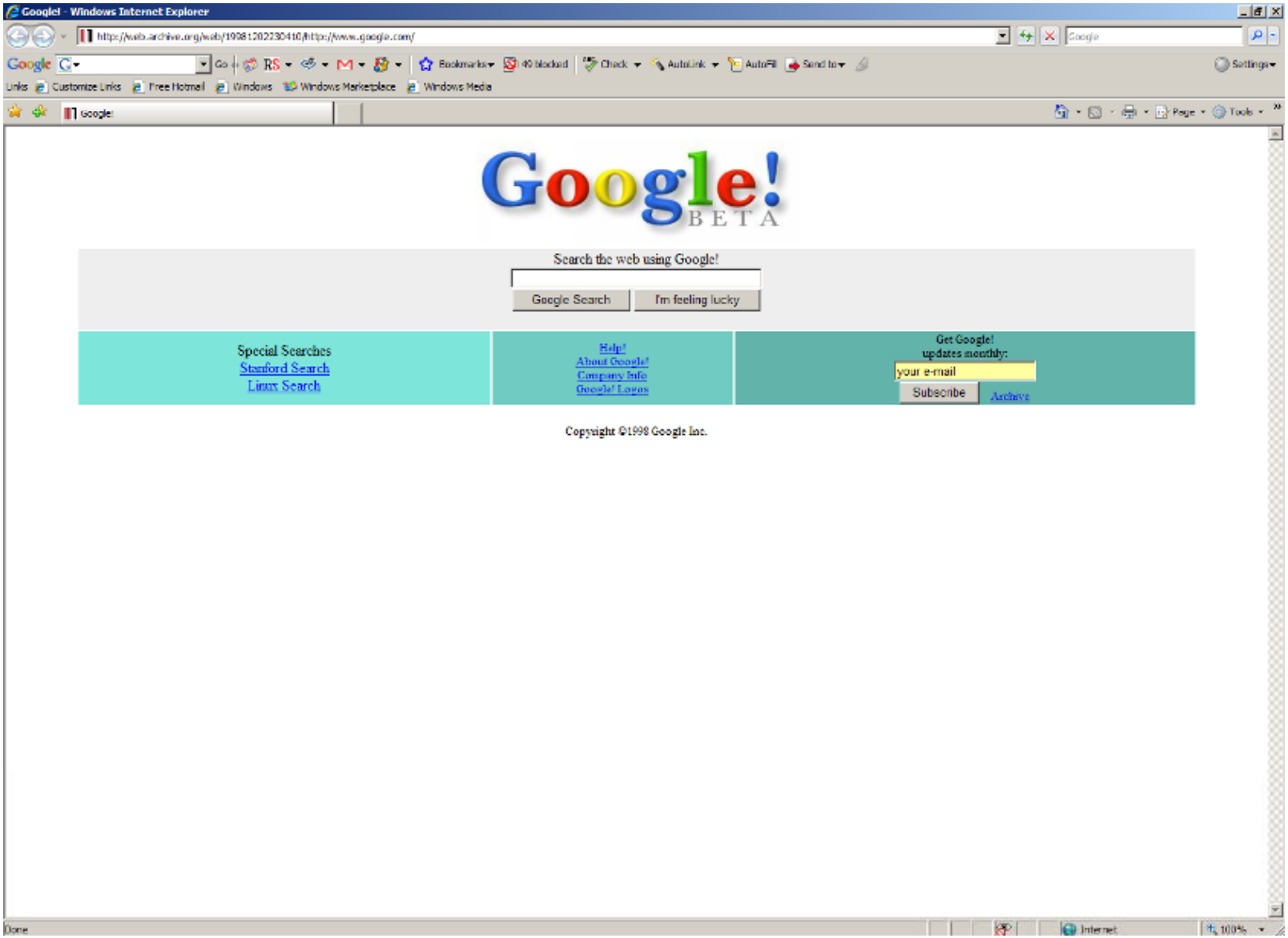
Internet Archive

- dedicated to building and maintaining a free and openly accessible online digital library, including an archive of the Web.

Internet Archive's WayBack Machine



Google - 1998


Google! - Windows Internet Explorer
Google!
BETA
Search the web using Google!
Google Search
I'm feeling lucky
Special Searches
Stanford Search
Linux Search
Help!
About Google!
Company Info
Google! Logos
Get Google! updates monthly:
your e-mail
Subscribe
Archive
Copyright ©1998 Google Inc.

Nutch

- Apache's Open Source Search Engine
- Distributed
- Tested with 100M Pages

WebSphinx

- 1998-2002
- Single Machine
- Lots of Problems (Memory leaks, …)
- Reported to be very slow

Crawler4j

- Single Machine
- Should Easily Scale to 20M Pages
- Very Fast

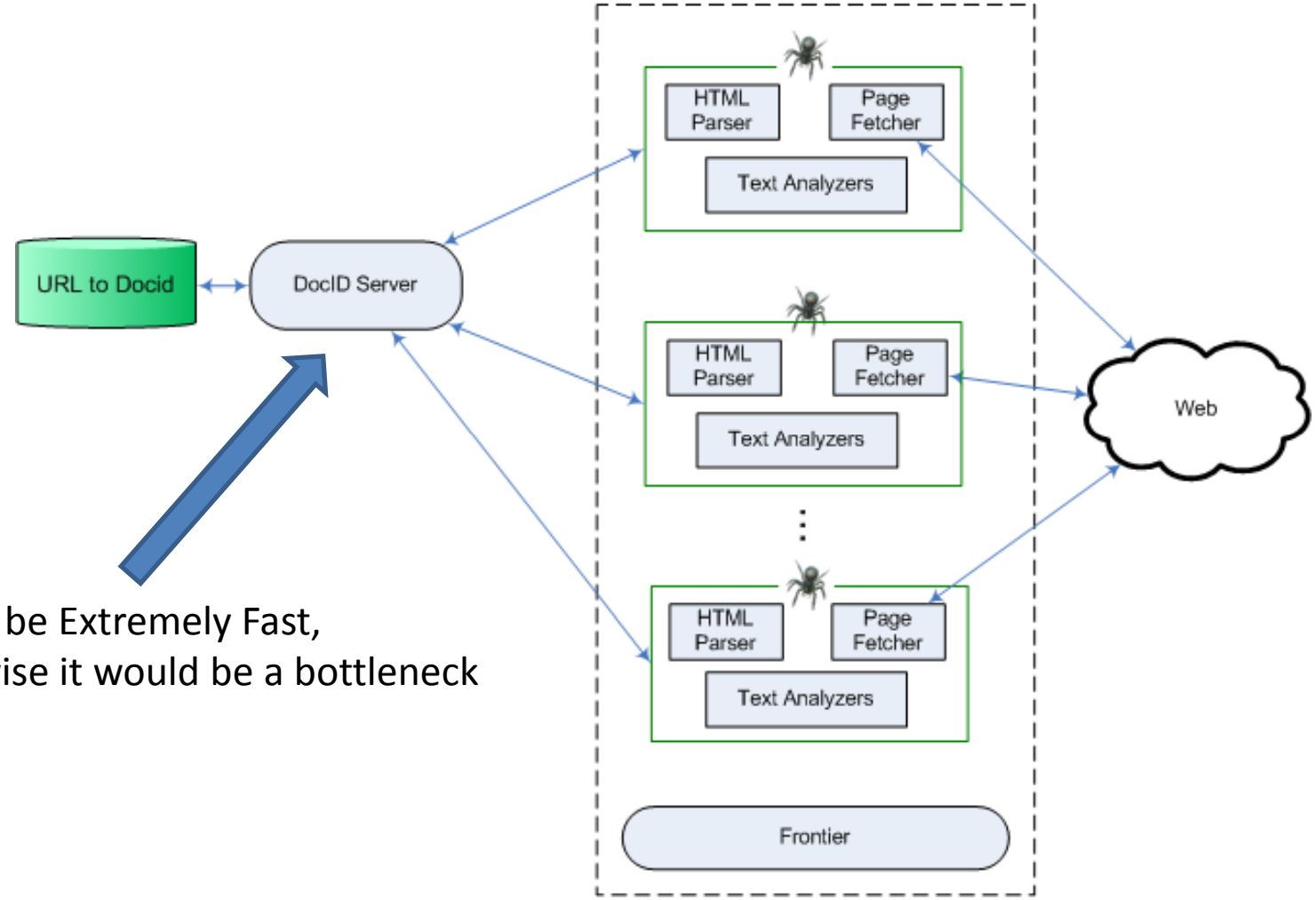
  Crawled and Processed the whole English Wikipedia in 10 hours (including time for extracting palindromes and storing link structure and text of articles).

What is a docid?

- A unique sequential integer value that uniquely identifies a Page/URL.

- Why use it?
  - Storing links:
    - “http://www.ics.uci.edu/”-”http://www.ics.uci.edu/about” (53 bytes)
    - 120-123 (8 bytes)
  - …

Docid Server



Should be Extremely Fast,
otherwise it would be a bottleneck

Docid Server

```
public  static  synchronized int getDocID(String URL) {
        if (there is any key-value pair for key = URL) {
                return value;
        } else {
                lastdocid++;
                put (URL, lastdocid) in storage;
                return lastdocid;
        }
}
```

Docid Server

- Key-value pairs are stored in a B+-tree data structure.
- Berkeley DB as the storage engine

Berkeley DB

- Unlike traditional database systems like MySQL and others, Berkeley DB comes in form of a jar file which is linked to the Java program and runs in the process space of the crawlers.
- No need for inter-process communication and waiting for context switch between processes.

- You can think of it as a large HashMap:

Crawler4j

- It is not polite
  - Does not respects robots.txt limitations
  - Does not limit number of requests sent to a host per second.
    - For example:
      - Wikipedia's policy does not allow bots to send requests faster than 1 request/second.
      - Crawler4j has a history of sending 200 requests/second
  - Introduces itself as a Firefox agent!
    - Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.4) Gecko/2008102920 Firefox/3.0.4
    - Compare with Google's user agent:
      - Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)

Crawler4j

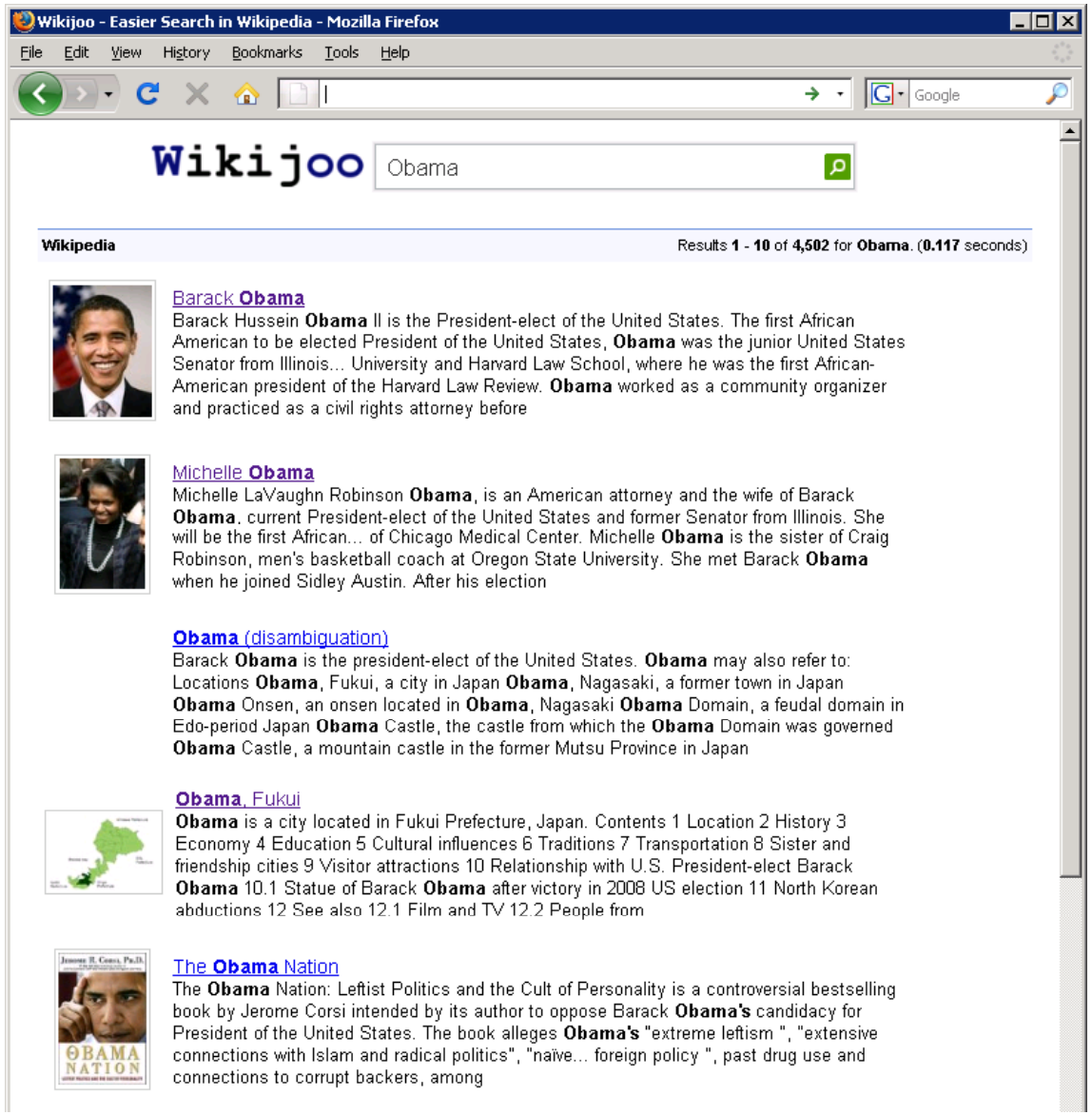
- Only Crawls Textual Content
  - Don’t try to download images and other media with it.

- Assumes that page is encoded in UTF-8 format.

Crawler4j

- There is another version of crawler4j which is:
  - Polite
  - Supports all types of content
  - Supports all encodings of text and automatically detects the encoding
  - Not open source ;)

Crawler4j – Crawler of Wikijoo



Your job?

```java
public class MyCrawler extends WebCrawler {

    Pattern filters = Pattern.compile(".*(\\.(css|js|bmp|gif|jpe?g"
            + "|png|tiff?|mid|mp2|mp3|mp4" + "|wav|avi|mov|mpeg|ram|m4v|pdf"
            + "|rm|smil|wmv|swf|wma|zip|rar|gz))$");

    public MyCrawler() {
    }

    public boolean shouldVisit(WebURL url) {
        String href = url.getURL().toLowerCase();
        if (filters.matcher(href).matches()) {
            return false;
        }
        if (href.startsWith("http://ics.uci.edu/")) {
            return true;
        }
        return false;
    }

    public void visit(Page page) {
        int docid = page.getWebURL().getDocid();
        String url = page.getWebURL().getURL();
        String title = page.getTitle();
        String text = page.getText();
        String html = page.getHTML();
        ArrayList<WebURL> links = page.getURLs();
    }
}
```

Your job?

```
public class Controller {
    public static void main(String[] args) throws Exception {
        CrawlController controller = new CrawlController("/extra/grad_space/yganjisa/crawl");
        controller.addSeed("http://ics.uci.edu/");
        controller.start(MyCrawler.class, 10);
    }
}
```

Crawler4j Objects

- Page
  - String html: getHTML()
  - String text: getText()
  - String title: getTitle()
  - WebURL url: getWebURL()
  - ArrayList<WebURL> urls: getURLs()
- WebURL
  - String url: getURL()
  - int docid: getDocid()

Assignment 03

- Feel free to use any crawler you like.

- You might want to filter pages which are not in the main namespace:
  - http://en.wikipedia.org/wiki/**Wikipedia:**Searching
  - http://en.wikipedia.org/wiki/**Category:**Linguistics
  - http://en.wikipedia.org/wiki/**Talk:**Main_Page
  - http://en.wikipedia.org/wiki/**Special:**Random
  - Image:, File:, Help:, Media:, …

- Set Maximum heap size for java:
  - java –Xmx1024M –cp .:crawler4j.jar ir.assignment03.Controller

Assignment 03

- Dump your partial results
  - For example, after processing each 5000 page write the results in a text file.

- Use *nohup* command on remote machines.
  - nohup java –Xmx1024M –cp .:crawler4j.jar ir.assignment03.Controller

Shell Scripts

run.sh:

```
#!/bin/bash
cp="."
for f in $(ls lib/*); do
    cp=$cp:$f
done
java -Xmx2048M -classpath $cp ir.assignment03.Controller
```

Available online: http://www.ics.uci.edu/~yganjisa/TA/

Shell Scripts

run-nohup.sh:

```
#!/bin/bash
cp="."
for f in $(ls lib/*); do
    cp=$cp:$f
done
nohup java -Xmx2048M -classpath $cp ir.assignment03.Controller > crawl-log.txt
```

Check logs:

```
tail -f crawl-log.txt
```

Available online: http://www.ics.uci.edu/~yganjisa/TA/

Tips

- Do not print on screen frequently.
- Your seeds should be accepted in the shouldVisit() function.
  - Pattern: http://en.wikipedia.org/wiki/*
  - Bad seed:
    - http://en.wikipedia.org/
  - Good seed:
    - http://en.wikipedia.org/wiki/Main_Page

Openlab

- From a Linux/MAC machine:
  - Connecting to shell:
    - ssh myicsid@openlab.ics.uci.edu
  - File transfer:
    - scp or other tools
- From a Windows machine:
  - Connecting to shell:
    - Download putty.exe
      - http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
  - File transfer:
    - WinSCP (http://winscp.net/)

QUESTIONS?